CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

An Adaptive Palmprint Enhancement and Feature Selection Method for Robust Identification

by

Ahmed Bilal Mehmood

A dissertation submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Faculty of Engineering
Department of Electrical Engineering

2023

An Adaptive Palmprint Enhancement and Feature Selection Method for Robust Identification

By

Ahmed Bilal Mehmood

(DCPE-153001)

Dr. Jose Valente de Oliveira, Senior Researcher

University of Lisboa, Portugal

(Foreign Evaluator 1)

Dr. Andrew Ware, Professor

University of South Wales, UK

(Foreign Evaluator 2)

Dr. Imtiaz Ahmad Taj

(Research Supervisor)

Dr. Noor Muhammad Khan

(Head, Department of Electrical Engineering)

Dr. Imtiaz Ahmad Taj

(Dean, Faculty of Engineering)

DEPARTMENT OF ELECTRICAL ENGINEERING

CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

ISLAMABAD

2023

To my parents, wife and children...

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY ISLAMABAD

Expressway, Kahuta Road, Zone-V, Islamabad
Phone:+92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: https://www.cust.edu.pk

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the dissertation, entitled "**An Adaptive Palmprint Enhancement and Feature Selection Method for Robust Identification**" was conducted under the supervision of **Dr. Imtiaz Ahmed Taj**. No part of this dissertation has been submitted anywhere else for any other degree. This dissertation is submitted to the **Department of Electrical Engineering, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Electrical Engineering.** The open defence of the dissertation was conducted on **November 17, 2023**.

Student Name :	Ahmed Bilal Mehmood (DCPE153001)	

The Examination Committee unanimously agrees to award PhD degree in the mentioned field.

Examination Committee :

(a)	External Examiner 1:	Dr. Asif ullah Khan Professor PIEAS, Islamabad	
(b)	External Examiner 2:	Dr. Usman Akram Professor CEME, NUST, Islamabad	
(c)	Internal Examiner :	Dr. Nadeem Anjum Associate Professor CUST, Islamabad	
	Supervisor Name :	Dr. Imtiaz Ahmad Taj Professor CUST, Islamabad	
	Name of HoD :	Dr. Noor Muhammad Khan Professor CUST, Islamabad	
	Name of Dean :	Dr. Imtiaz Ahmad Taj Professor CUST, Islamabad	

AUTHOR'S DECLARATION

I, **Ahmed Bilal Mehmood (Registration No. DCPE153001)**, hereby state that my dissertation titled, "**An Adaptive Palmprint Enhancement and Feature Selection Method for Robust Identification**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/ world.

At any time, if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my PhD Degree.

(Ahmed Bilal Mehmood)

Dated: 17 November, 2023

Registration No : DCPE153001

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the dissertation titled "**An Adaptive Palmprint Enhancement and Feature Selection Method for Robust Identification**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete dissertation has been written by me.

I understand the zero-tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled dissertation declare that no portion of my dissertation has been plagiarized and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled dissertation even after award of PhD Degree, the University reserves the right to withdraw/ revoke my PhD degree and that HEC and the University have the right to publish my name on the HEC/ University Website on which names of students are placed who submitted plagiarized dissertation.

(**Ahmed Bilal Mehmood**)

Dated: 17 November, 2023

Registration No : DCPE153001

List of Publications

It is certified that following publication(s) have been made out of the research work that has been carried out for this dissertation:-

1. **Mehmood, A. B.**, Taj, I. A., Ghafoor, M. (2023). Palmprint enhancement network (PEN) for robust identification. Multimedia Tools and Applications, 1-28. DOI: *https://doi.org/10.1007/s11042-023-16043-z*

2. **Mehmood, A. B.**, Taj, I. A., Ghafoor, M. (2023). A Minutiae Selection Algorithm (MSA) for efficient palmprint matching using Histograms of Differences (HoDs). Expert Systems with Applications, 120734. DOI: *https://doi.org/10.1016/j.eswa.2023.120734*

(Ahmed Bilal Mehmood)
Registration No: DCPE 153001

Acknowledgement

I would like to thank Dr Imtiaz Ahmad Taj (my supervisor) and the whole team of Vision and Pattern Recognition Group (VisPrs) for their kind support and guidance. I would also to thank Dr Mubeen Ghafoor for collaborating with me during the course of my research, and Dr Jianjeng Feng for making THUPALMLAB dataset public. I am also grateful to my family for their prayers and continuous support.

(Ahmed Bilal Mehmood)

Abstract

The amount of information provided by high resolution palmprints and the fact that they have considerable forensic value makes them a preferred biometric choice for large-scale identification systems. In high resolution palmprints, extraction of reliable features (minutiae) for identification is still a challenging task especially because most of the palmprints found in the real world, e.g., in crime scenes, are of poor quality. These palmprints suffer from multiple degradations, namely, poor contrast, background textures, occlusions, poor ridge structure, false minutiae, etc. Due to these degradations, palmprints cannot be directly used for identification and need to go through various pre-processing steps. Keeping these challenges in view, this thesis presents contributions in three areas.

An efficient frequency-domain ROI segmentation method is proposed that uses a bandpass filter to isolate only those regions of the image that contain palmar ridge patterns. Background textures and noise usually belong to low and high frequencies and are effectively filtered out. Proposed method is independent of variations in image intensity and contrast. Results show that the proposed frequency-domain segmentation method out performs previously proposed texture-based segmentation methods. Proposed ROI segmentation helps in limiting the subsequent processing including enhancement, feature extraction and matching to only the valid regions of an otherwise very large image.

A two-step deep learning-based palmprint enhancement network (*PEN*) is proposed that is able to convert considerably large patches of palmprints directly into corresponding enhanced patches by accentuating the palm ridge structure. In the first step, a classification Convolutional Neural Network (Cnet) predicts dominant ridge orientation in a patch of palmprint. Guided by this orientation prediction, the patch passes through an image-to-image regression network (Rnet) which converts the patch to its corresponding enhanced patch. All enhanced patches are joined together to produce a complete enhanced palmprint. Both Cnet and Rnet are trained separately. Cnet is a fine-tuned version of alexnet, while Rnet is a 4-layer deep network that is designed and trained from scratch. Results show that when compared to state-of-the-art methods that use classical pixel-wise contextual

filtering methods either in the spatial or frequency domain, the proposed patch-wise enhancement method is found to be more responsive to abruptly changing ridge orientation and frequency in the palmprint. This is due to the fact that Rnet has a sufficient number of kernels and depth to accommodate abrupt changes in the ridge patterns. The performance of Rnet for palmprint enhancement is compared with two popular deep learning paradigms, i.e., U-net and Resnet. Due to a relatively simpler design and training procedure, the proposed method achieves higher accuracy, both in the enhancement and identification of palmprints.

Lastly, an intuitive minutiae selection algorithm (*MSA*) is proposed that reduces the number of minutiae required to match palmprints. This is important because apart from the poor quality of high resolution palmprints, another problem encountered during matching is the high computational overhead incurred due to the extraction of a large number of minutiae from the palmprint. Since most of these minutiae are false, they prove detrimental for identification accuracy as well. This simple histogram-based iterative algorithm utilizes only the basic properties of minutiae, i.e., (x, y, θ) to shortlist a subset of best minutiae candidates for matching. This provides the dual benefit of: 1) a reduced number of minutiae matches between two palmprints, and 2) improved matching accuracy through the elimination of false minutiae.

All the results are acquired on the most popular and challenging high resolution dataset that has been used in all state-of-the-art studies.

Contents

List of Figures

List of Tables

Abbreviations

PIN	Personal identification number
FAR	False acceptance rate
FRR	False rejection rate
ppi	Pixels per inch
CCD	Charged couple device
ROI	Region of interest
CAGR	Compound annual growth rate
CNN	Convolutional neural network
MFRAT	Modified finite radon transfom
DoG	Difference of gaussian
PEN	Palmprint enhancement network
SIFT	Scale invariant feature transform
LBP	Local bit pattern
LDP	Local direction pattern
DGLBP	Directional gradient with LBP
MSA	Minutiae Selection Algorithm
HoD	Histogram of Differences

Symbols

∇_x	Image gradient in x-direction
∇_y	Image gradient in y-direction
A^l	Output of convolutional layer l
G^l	ReLU activation of lth layer
W_k^l	Weights of k kernels in lth layer
B_k^l	Biases of k kernels in lth layer
$L(W)$	Mean squared error

Chapter 1

Introduction

1.1 Overview

The measurement and analysis of human physical characteristics is called Biometrics. The use of biometrics for human identification has been in practice for centuries. The last two decades in particular have seen substantial increases and innovations in the use of biometrics. Although other digital identities such as passwords, Personal identification numbers (PIN), and smart cards have been mainstream in the recent past, these means of identification have failed to provide an adequate sense of security to users. This is because digital IDs can be stolen or faked easily. On the other hand, biometric identities provide a higher level of security and accuracy.

In today's modern era, human interaction and businesses are quickly shifting to digital platforms. COVID-19 became a catalyst in the mass adoption of this digital culture resulting in consumers embracing biometrics with an increased level of comfort for routine activities such as checking in at airports, online payments, etc. Security concerns have also escalated among the public as well as law enforcement agencies. The use of biometrics such as fingerprints, palmprints, face, iris, etc. is not new but modern biometric trends are moving beyond traditional practices. Biometric technologies have quickly replaced digital IDs such as passwords and PINs and have potentially prevented unauthorized access to:

- Cell phones
- ATM machines
- Computers
- Offices
- Airports

This sharp increase in the utility of biometrics has encouraged researchers to improve the efficiency and accuracy of biometric systems. The main objective of a biometric system is to automatically recognize the identity of a person using physiological or behavioral characteristics or features. In order to achieve these goals, reliable features need to be employed that exhibit the following characteristics:

- Uniqueness: Selected features should be able to uniquely identify a person
- Universality: Selected features should be present in all persons
- Persistence: Selected features should remain persistent during the lifetime of a person
- Easy acquisition: Selected features should be extractable using simple methods and instruments
- User comfort: Selected features should be acquired with minimum inconvenience to persons

1.2 Historical Background

Search for appropriate physiological traits that can precisely discriminate one individual from other is not new. Interest in this field grew rapidly during the Age of Enlightenment in Europe when every field of life started getting influenced by scientific research. Police investigative procedures also started incorporating scientific methods to improve the efficiency of their criminal identification systems.

First dedicated effort to find out reliable human physiological features that do not deform with age and are persistent under varying climates was made by Alphonso Bertillon in the late nineteenth century. He created a human identification system taking into account body measurements, identification marks, and other physical characteristics. He called it "anthropometry". He proposed that body dimensions are persistent enough to last a complete lifetime and can be used to discriminate one person from other. Although it was a good starting point for biometric identification, it had two major flaws. Firstly, multiple individuals can have similar body dimensions. Secondly, in case of criminal investigation, no evidence will be left behind once the criminal leaves the crime scene.

This led to the use of fingerprints as the prime biometric identification system. The use of fingerprints dates back to 300 BCE in ancient China. However, it was in India in the nineteenth century CE, that William Herschel proposed the use of fingerprints and palmprints for registering citizens owing to the strong discriminative properties of both. He used fingerprints and palmprints on contract deeds in order to verify the authenticity of documents (see Figure 1.1). He also used fingerprints and sometimes complete palmprints to register all government employees in order to stop the collection of pensions by an employee's relatives after the employee's death. A little later, Dr. Henry Faulds, who was a surgeon in Tokyo, became the first researcher to publish his research on the effectiveness of human identification based on fingerprints.

Later, it was conclusively proven by Galton [1] that fingerprints were unique for every individual. Although there is a small amount of similarity between the fingerprints of family members, but it is not to the point that one family member can be mistaken for someone else. He also argued that fingerprints were persistent during the lifetime.

Based on Galton's research, Juan Vucetich, an Argentinian police officer became the first person to create a database of fingerprints. This database was later incorporated into the Argentinian crime investigation department. Juan Vucetich became the first police officer in history to solve a crime using the fingerprints left at the crime scene. In later years, acquiring of fingerprints for human identification more popular.

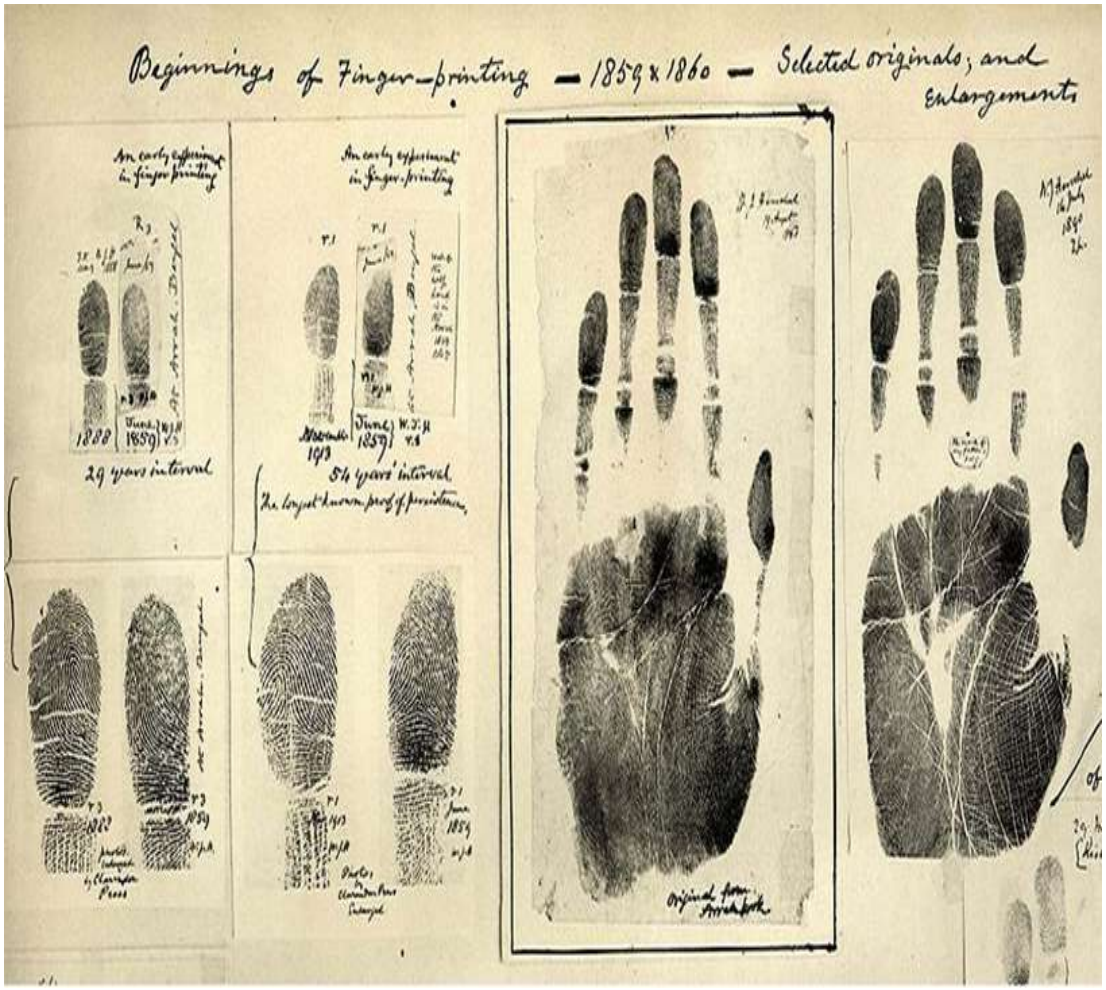

FIGURE 1.1: Fingerprints taken by William Herschel 1859/1860 [2]

Since then, the study of biometrics has come a long way. Search for suitable physical traits to verify a person's identity has opened multiple research areas encompassing features extracted from the iris, face, hand geometry, finger and palm veins, etc. Modern instruments and sensing techniques have enabled researchers to use behavioral characteristics of people for identification as well. The following section briefly introduces some of the currently popular biometric systems. Regardless of the biometric modality being used, the general architecture of biometric systems has remained the same. There is an enrollment stage in which suitable biometric trait of all candidates is acquired, processed, and stored in a database as a template. In the online stage, acquired biometrics are processed and matched with templates of all candidates in the database. Figure 1.2 illustrates the architecture of a fingerprint-based biometric system.

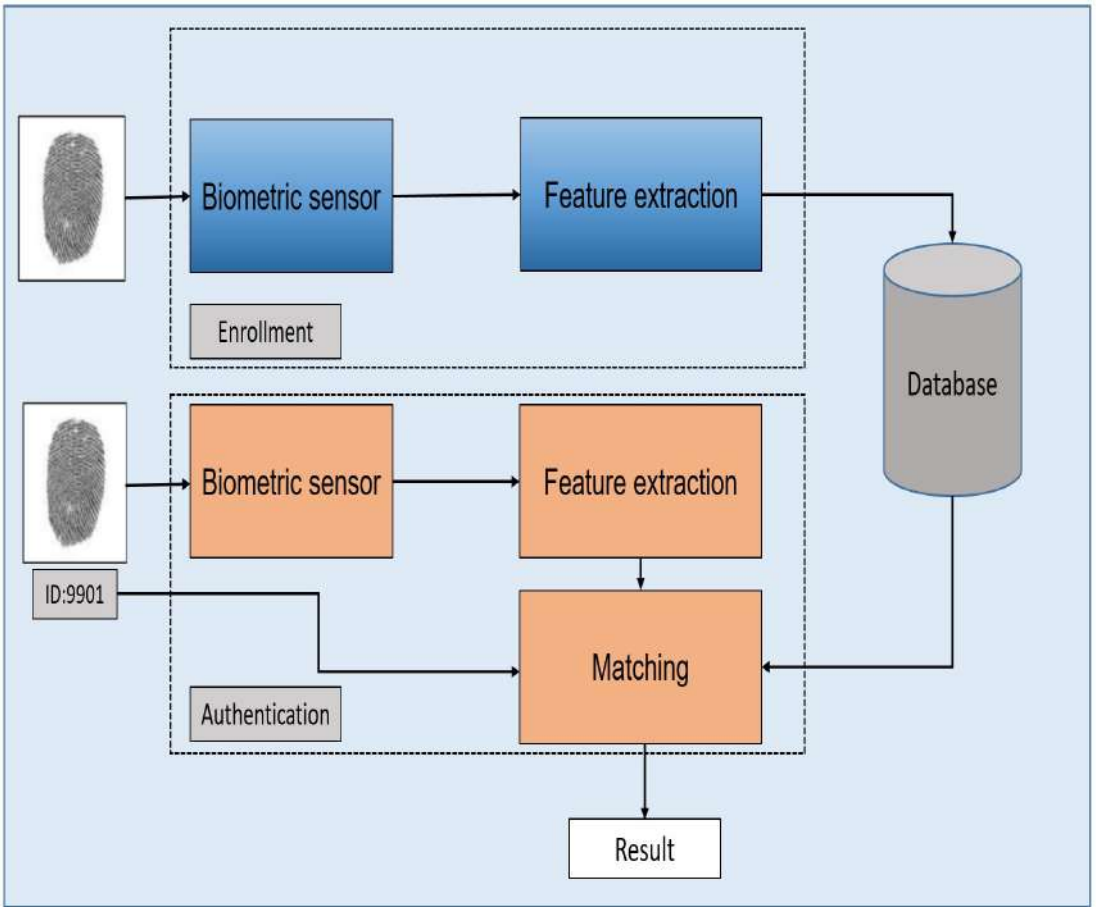

FIGURE 1.2: General architecture of a biometric system

1.3 Types of Biometric Systems

Biometric systems can be categorized into Physiological or Behavioral biometric systems (Figure 1.3). Physiological biometrics use the physical characteristics of a person that show minimal change during the lifetime of a person. These include fingerprints, palmprints, iris, face and hand geometry, etc. On the other hand, behavioral biometrics identify a person by identifying patterns in a person's activities. These include gait analysis, keystroke dynamics, speech identification, signature analysis, etc. Physiological biometric systems are more accurate but Physiological biometric identities can be stolen. Behavioral biometric systems don't need any special sensors for acquisition and are therefore less costly. Also, behaviors are integral to a person and hence identities based on human behaviors cannot be stolen. In the following section, popular physiological and behavioral biometrics are listed.

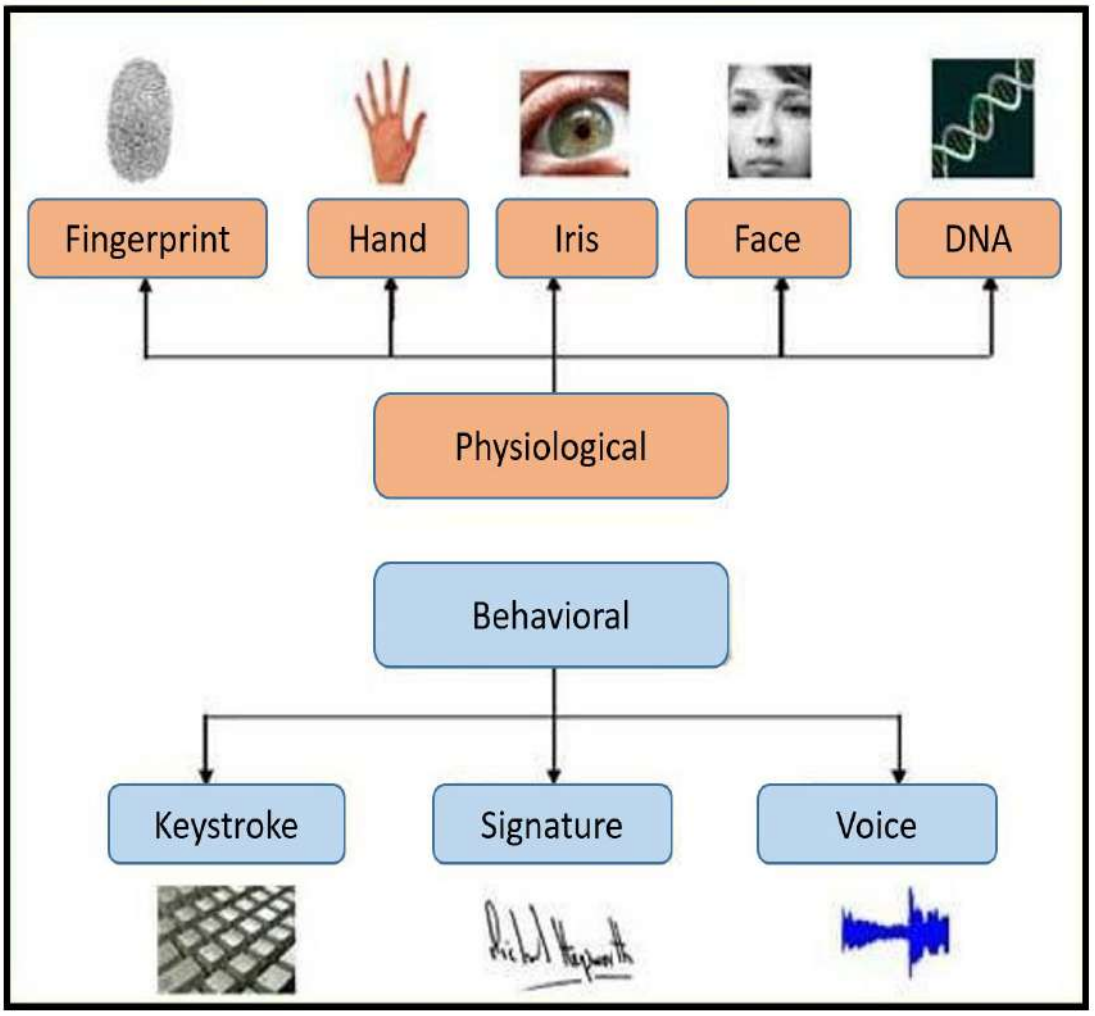

FIGURE 1.3: Categories of Biometric Systems: Including the physiological and relatively new behavioral biometrics[3]

1.3.1 Physiological Systems

1.3.1.1 Fingerprints

One of the oldest methods of verifying the identity of a person. Features contained within a fingerprint are reliable, unique, and persistent during the lifetime of a person. Fingerprints contain a peculiar pattern of alternating ridges and valleys (Figure 1.4). In fingerprints, ridge lines are of utmost importance as they help in uniquely identifying a person. The ridge structure exhibits discontinuities which are either ridge endings or bifurcations. These discontinuities are exploited to uniquely identify a person. Sensors used for fingerprint scanning are commonplace and provide a level of convenience to users.

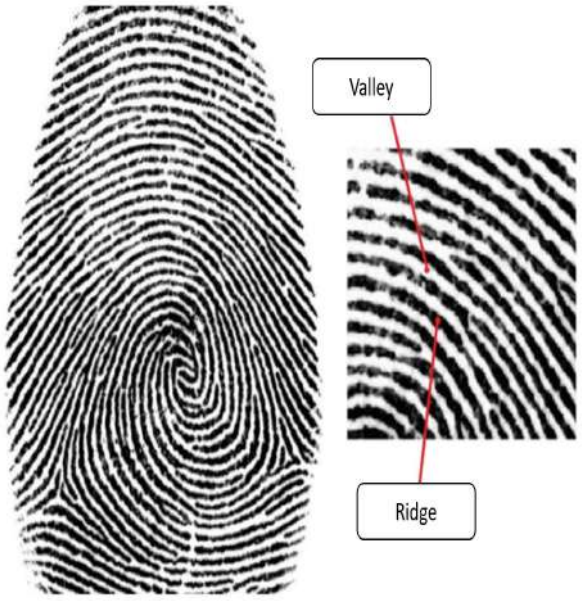

FIGURE 1.4: Fingerprint sample

1.3.1.2 Iris

Iris is the circular portion between the retina and the lens of the eye illustrated in Figure 1.5. It provides highly stable features for verifying the identity of a person. Iris is first segmented out and broken down into blocks and subsequently converted using numbers which become the unique template of a person. Iris provides exceptional identification accuracy but iris-based biometric systems have few downsides. Firstly, sensors required for iris scanning are expensive. Secondly, the iris cannot be scanned from distances more than two meters. This means users have to experience inconvenience while dealing with iris-based systems.

1.3.1.3 Face

Non-invasive nature of facial biometric systems has made the face a popular biometric choice for large-scale identification systems. Face recognition systems measure distances between various features on the face. These include:

- Distance between eyes, nose and mouth, forehead and chin
- Contours of lip, ears, and chin

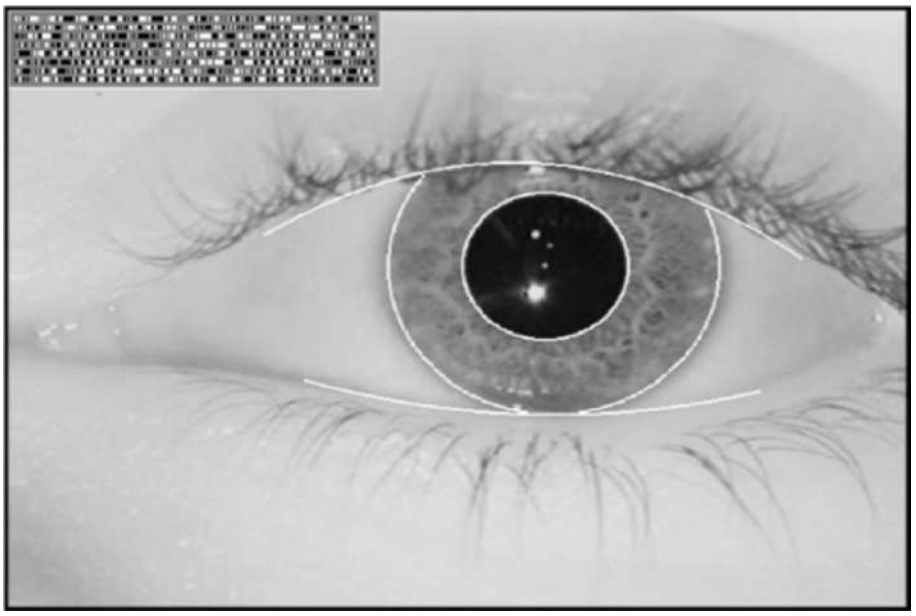

FIGURE 1.5: Iris sample

- Shape of cheekbones
- Depth of eye sockets

Face recognition systems have seen high demand in access control systems as well as forensics. However, face recognition systems have certain limitations. Limitations mainly arise from aging effects on the face. The use of face recognition systems in uncontrolled environments is also limited due to challenges arising from pose variations and background illumination, etc.

1.3.1.4 DNA

Deoxyribonucleic acid (DNA) is present in all cellular material like blood, hair, skin, etc. but it is usually acquired from the inside of the mouth using a cotton swab. Later, the DNA sample is cut into smaller segments and smaller strands of DNA are separated from larger ones. The accuracy of DNA profiling depends on the number of segments analyzed. Although DNA provides an accurate and stable means of identifying a person, the time required for its extraction and post-processing renders it an unsuitable choice for large-scale identification systems. Secondly, enrollment of persons in a DNA-based database is a cumbersome process that is likely to put off most people.

1.3.1.5 Palmprints

Just like fingerprints, palmprints exhibit a peculiar pattern of alternating ridges and valleys. They provide much more information as compared to fingerprints and provide additional features as well. These additional features include principal lines and creases illustrated in Figure 1.6. Palmprints have remarkable forensic value as they are more likely to be found on surfaces as compared to fingerprints. Furthermore, it is much more difficult to fake palmprints as compared to fingerprints. Like fingerprints, palmprints have peculiar epidermal ridge patterns which

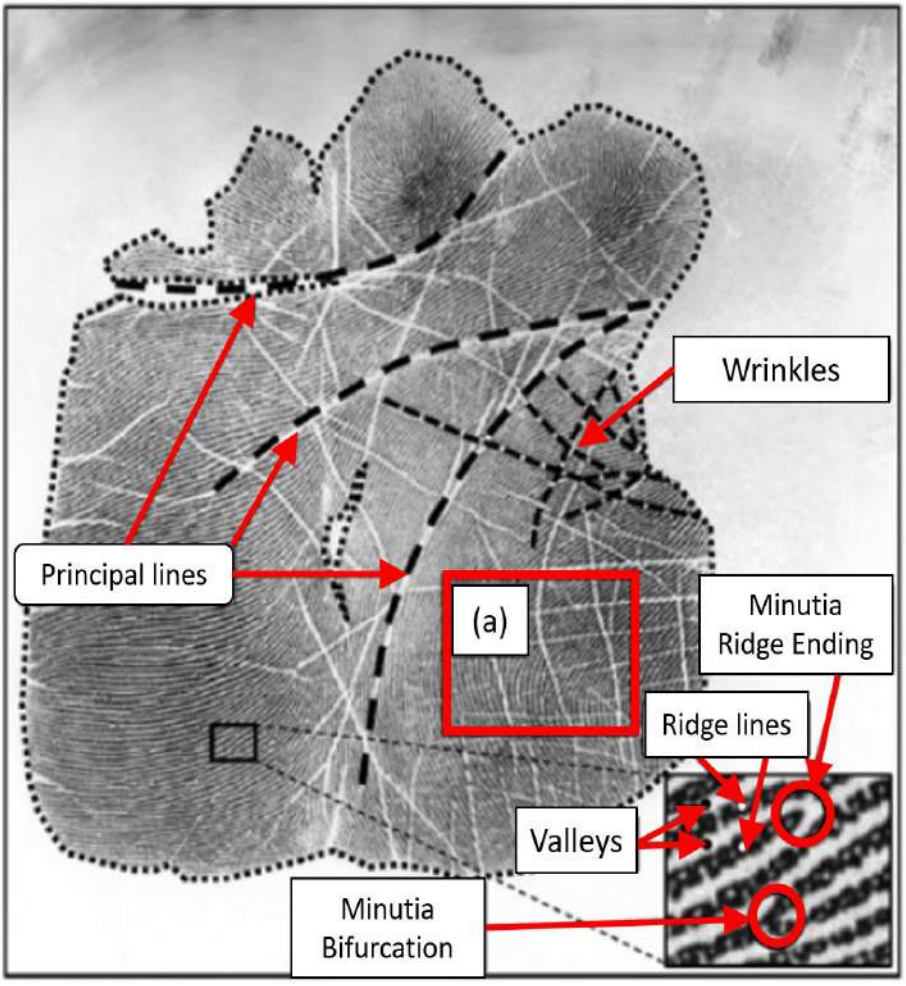

FIGURE 1.6: Palmprint sample showing all intrinsic features: (a) region with creases

are exploited to identify humans. Based on the properties of these ridges a variety of palmprint authentication methods have been developed, e.g. correlation-based, ridge pattern-based, or minutiae-based. The most accurate of these are minutiae-based methods. Since the focus of this thesis is on palmprints, all components of

a palmprint recognition system are discussed in the proceeding chapters.

Minutiae are minute details found in ridge patterns found in fingerprints or palmprints. Most notable of these are ridge endings and ridge bifurcations. A ridge ending is a point where a ridge ends and ridge bifurcation is a point where a single ridge splits into two. Both types are illustrated in Figure 1.7. Minutiae points have other types too, but all of those can be constructed through combinations of the two types mentioned above.

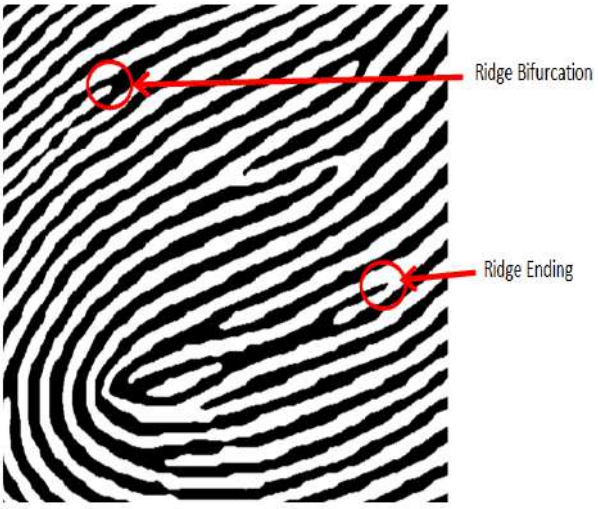

FIGURE 1.7: Portion of a binarized palmprint showing minutiae points (ridge endings and bifurcations). Black lines are ridges, White lines are valleys [4].

1.3.2 Behavioral Systems

1.3.2.1 Keystroke Dynamics

Keystroke dynamics recognize patterns in typing behaviors of a person. These patterns can include the speed of typing, use of the shift key or use of backspace for mistakes, etc. Any anomaly in these patterns is identified and reported. Unlike physiological biometric systems, keystroke dynamics do not require an active scan of a person. Furthermore, they offer continuous validation of a person's identity rather than just once. In case any anomaly is detected in usual patterns attributed to a person, the system can ask for additional validation.

1.3.2.2 Signatures

A person's signature is a modality that develops over time and becomes unique to a person. With the shifting of most human processes to digital formats, most of the paperwork has also taken up digital form. Documents are scanned and sent to recipients in digital forms. Since documents have legal value, the main challenge in this process is the authentication of signatures on a document. Verification of signatures can be done in a static (offline) or dynamic (online) fashion. In the static model, the scanned image of a signature is analyzed based on various aspects of its shape and size and compared with templates available in the database. In the dynamic model, the live signatures of a person are acquired on a digital pad. Apart from shape and size, additional features are verified that include but are not limited to the speed of signatures, pressure at each point, etc. illustrated in Figure 1.8.

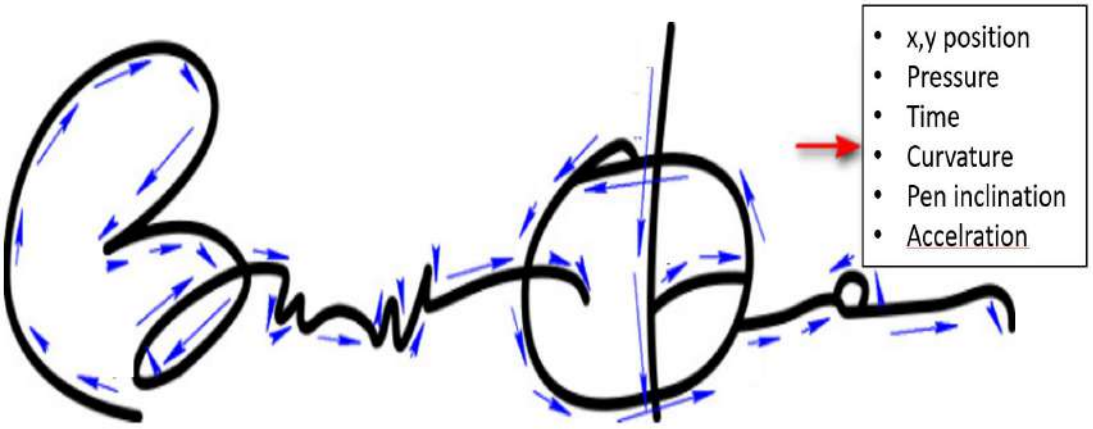

FIGURE 1.8: Example of features available in dynamic signature verification

1.3.2.3 Voice

Each person's voice has unique characteristics that distinguish them from other people. The shape of the mouth and throat, pitch, and speaking patterns all dictate the output voice of a person. The input voice of a speaker is processed

and suitable features from the voice are extracted and compared with one or more samples available in the database. Based on similarity, the identity of a person is validated.

1.4 Comparison of Biometrics

Each biometric technology comes with its own set of challenges and benefits. The choice of biometrics is usually dependent on the application. Choosing accuracy of identification as a metric of success seems an appropriate decision but sometimes systems that give high accuracy are not acceptable to users. For example, the iris is exceptional in terms of accuracy and longevity but users find iris-based biometric systems to be intrusive. Similarly, DNA-based systems are also highly accurate but collecting DNA samples and enrolling them in the database is a tiresome process. Similarly, palm or finger vein systems are highly accurate but need specialized scanners which are not commonplace.

Selection of an appropriate biometric is a trade-off between different considerations which include Ease of Use, user acceptability, accuracy, challenges, susceptibility to errors, Longevity, etc. A comparison of different biometrics on the above-mentioned aspects is presented in Figure 1.9. The table shows that while some biometric technologies are more accurate, they can not be chosen for mass adoption due to low user acceptance or collectibility of samples. On the other hand, less accurate technologies are widely acceptable to users and are easily adopted by the masses.

1.5 Performance Evaluation of Biometric Systems

The performance of biometric systems is usually measured in terms of accuracy with which they can classify test (query) samples as match or no-match with templates saved in the database. Biometric systems do not give an absolute answer

Characteristics	Fingerprints	Hand Geometry	Retina	Iris	Face	Signature	Voice
Easy of Use	high	high	Low	Medium	Medium	High	High
Error Incidence	Dryness, dirt, age	Hand injury, age	Glasses	Lighting	Lighting, age, glasses, hair	Changing signature	Noise, colds
Accuracy	High	High	Very high	Very high	High	High	High
User Acceptance	Medium	Medium	Medium	Medium	Medium	High	high
Long Term Stability	High	Medium	high	high	Medium	Medium	Medium

FIGURE 1.9: Comparison of Biometric technologies [3]

to whether a test sample is matched or not. Instead, the similarity score between the test (query) sample and templates stored in the database is calculated and an acceptability threshold on the similarity score decides whether it matches a template or not. Results provided by the system can also be classified as inconclusive if the overall similarity scores calculated by the system are not satisfactory.
The following parameters are most commonly used in evaluating the performance of biometric systems:

- **False Acceptance Rate (FAR)**: Rate of wrongful classification of a non-genuine sample as genuine
- **False Rejection Rate (FAR)**: Rate of wrongful classification of a genuine sample as non-genuine
- **Equal Error Rate (EER)**: Rate where FAR and FRR are equal (Figure 1.10)

FAR and FRR are interdependent and it is not possible to reduce one rate without reducing the other. That's why the most effective biometric systems are the ones with low FAR and FRR resulting in low EER. Acceptability threshold t in Figure 1.10 can be varied. Selecting a higher threshold means lower FAR but higher FRR and vice versa.

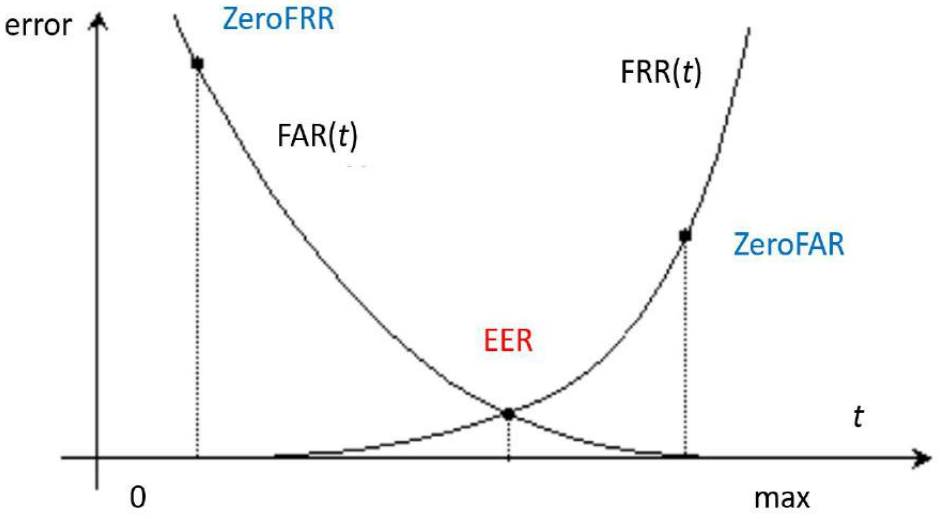

FIGURE 1.10: EER denotes when FAR=FRR, ZeroFAR denotes FRR when FAR=0, ZeroFRR denotes FAR when FRR=0, t is the acceptability threshold

1.6 Use of Palmprints for Identification

Palmprints show persistence across time and climates. Palmprints are unique to a subject and provide a sound foundation for the correct identification of individuals. Even identical twins have different finger and palmprints [5]. As compared to fingerprints which contain only minutiae-based features, palmprints contain multiple types of intrinsic features that can be used for identification. In addition to minutiae-based features, palmprints contain principal lines and wrinkles that can also be utilized to uniquely identify a subject. In order to understand the potential of palmprints, it is important to understand the physiology of the human hand. Figure 1.11 illustrates various features in a palm.

Palm features illustrated in Figure 1.11 are formed during the embryonic stages of a fetus. Features contained within a palm can be categorized as follows:

- Major Flexion Creases (Principal Lines): These are the major lines easily visible in a palm to the naked eye. They follow a strict spatial pattern and this pattern does not change at any point in time. These are the Distal Line, Proximal Line, Thenar or Radial Line commonly referred to as Heart Line, Head Line, and Life Line respectively. There is another type of crease, known as minor flexion crease. These are mostly found in the thenar region

of the palm and introduce discontinuities in local ridge patterns. They can introduce a lot of errors during the enhancement or matching of palmprints.

- Ridges: Also known as friction ridges, help a hand in improving its grip on objects. These structures are not easily visible to the naked eye and need high-resolution scanners/ cameras to become apparent.
- Minutia: As described earlier, minutiae are ridge endings or bifurcations.
- Pores: Pores are visible at extremely high resolutions (ppi>1000)

Palmprint features listed above are obtained at different image resolutions. Some features are available at low resolution while others are available at high resolution. Low resolution features are called Level 1 features and are obtained at a resolution below or equal to 100 ppi, high resolution features are called Level 2 features which are obtained at image resolutions above 400 ppi, and then there

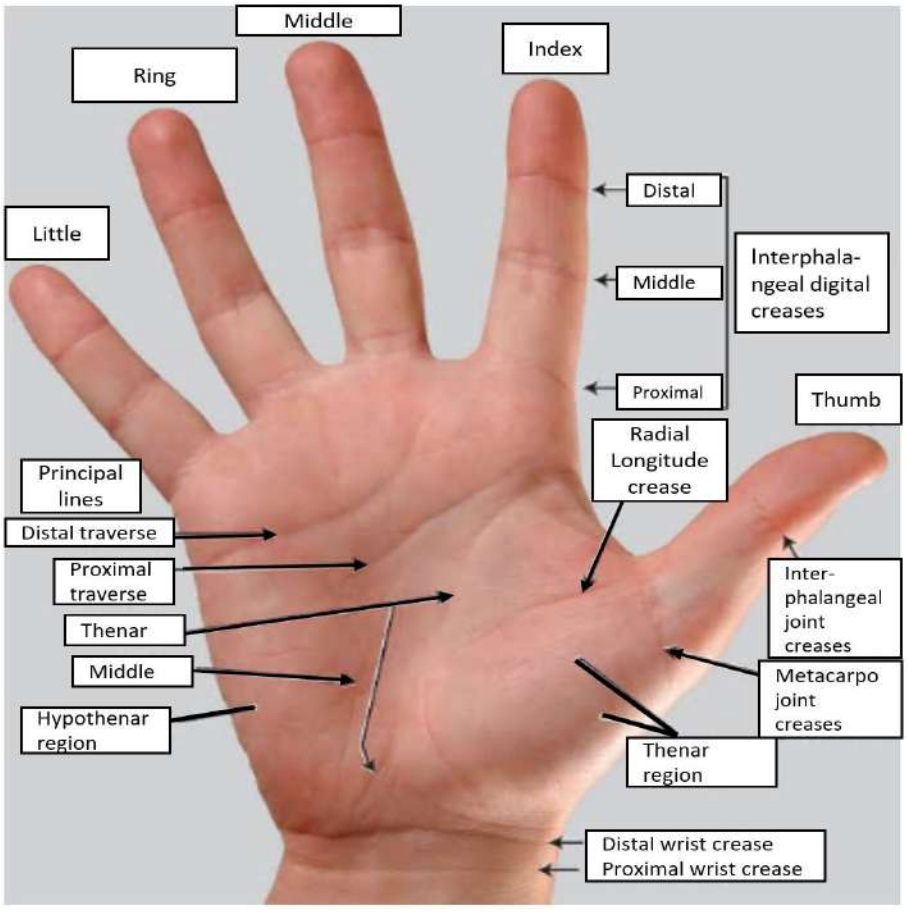

FIGURE 1.11: Physiology of human hand [6].

are Level 3 features which are obtained at ultra-high resolutions equal to or above 1000 ppi using specialized sensors. Level 1, Level 2, and Level 3 features are listed in Table 1.1.

	Level 1 (100 ppi)	**Level 2 (400-500 ppi)**	**Level 3 (1000 ppi)**
Features	Principal Lines	Minutia	Pores
	Wrinkles	Minor Creases	Ridge Width
	Texture	Ridges	
		Valleys	

TABLE 1.1: Palmprint Features Available at Different Resolutions.

Figure 1.12 illustrates these multi-level features thoroughly. Based on the image acquisition method and resolution being used, appropriate features can be selected for a specific application. This freedom in the choice of features is not available in other biometric systems such as fingerprints, iris, etc. Apart from the features, it is important to know the different regions of a palm, namely, interdigital, thenar and hypothenar. As will become apparent in subsequent chapters, different regions in a palm respond differently to enhancement and matching algorithms.

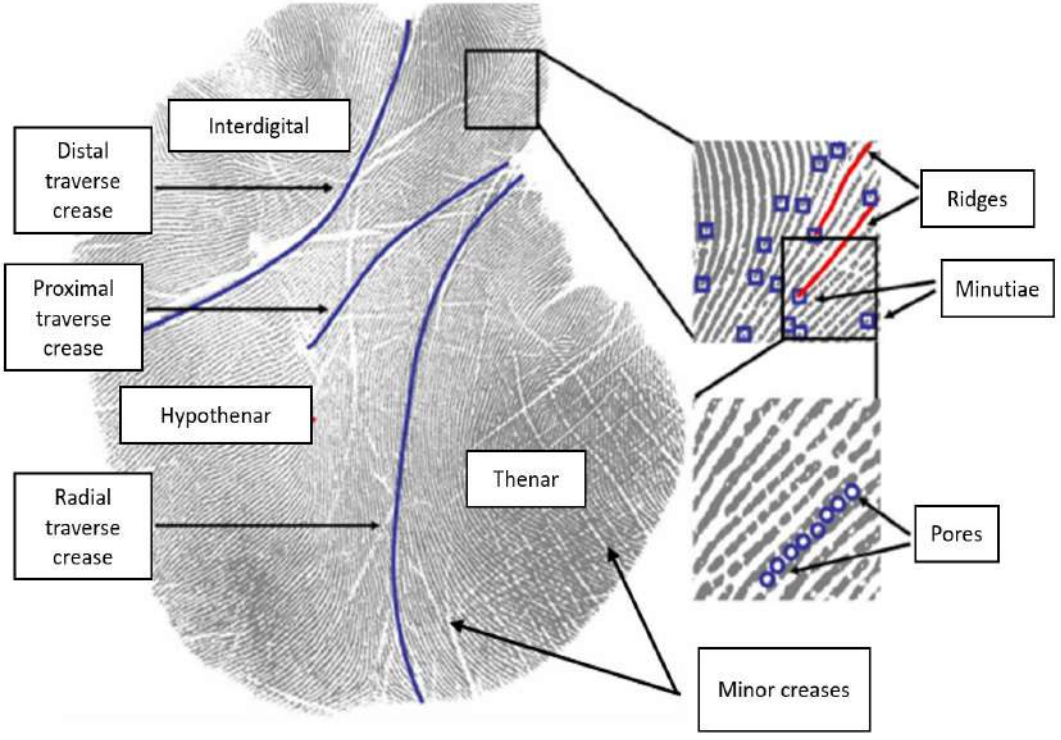

FIGURE 1.12: Level 1 Features (Distal, Proximal and Radial Transverse Creases), Level 2 Features (Ridges, Minutia), Level 3 Features (Pores) [7]

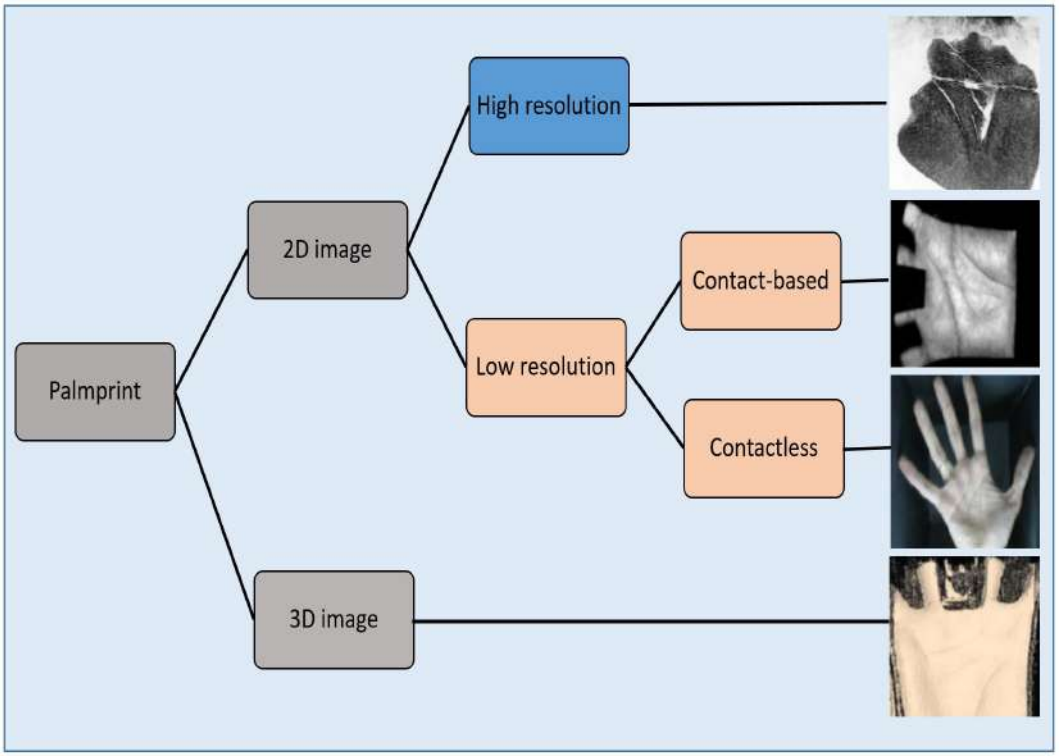

FIGURE 1.13: Palmprint categories: based on dimensions and resolutions of image [8]

Based on the type and quality of images, palmprint identification systems can be divided into high resolution and low resolution systems. Low resolution systems can be further divided into contact-based or contact-less systems based on the method of image acquisition. In contact-based methods, the palm is in contact with the surface of the image sensor, while in contact-less methods, the palm is not in contact with any sensor. Palmprints can also be categorized on the basis of the dimensions of the image, i.e., 2D (two-dimensional) or 3D (three-dimensional) systems. All categories are illustrated in Figure 1.13.

Low resolution systems provide a certain level of convenience due to easy and inexpensive image acquisition methods like a smartphone or a CCD camera. But low resolution systems make use of Level 1 features which provide low to middle-level security and feature templates stored in a database is vulnerable to spoofing attacks. As a result, feature templates have to be protected using encryption techniques [9], or palmprint features are fused with other biometrics such as fingerprints, iris, etc. to form a multi-modal identification system [10]. High resolution systems make use of Level 2 features (ridges/ valleys and minutiae) which

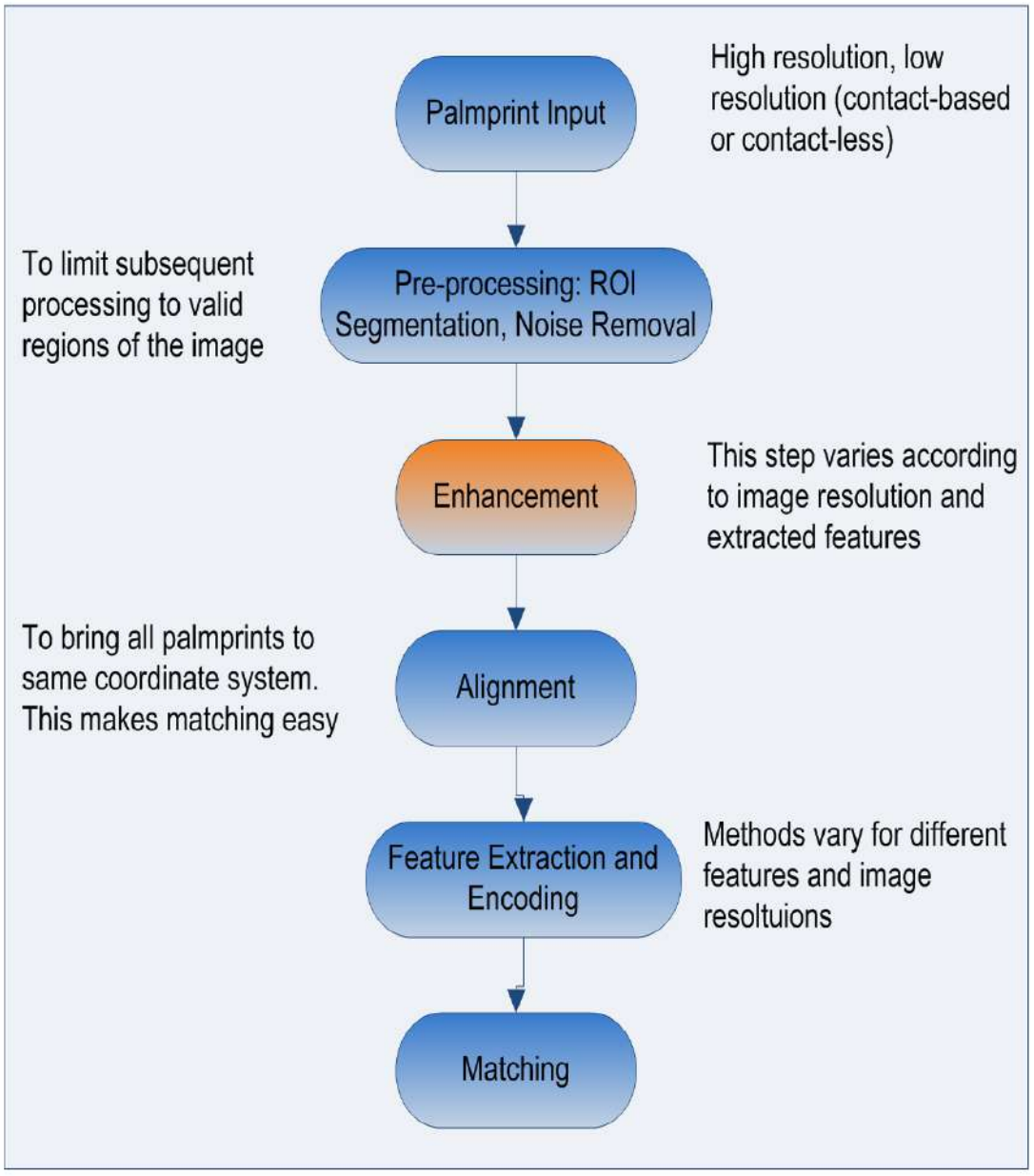

FIGURE 1.14: Generic architecture of a palmprint identification system

are considered the most reliable and have the advantage of providing latent-to-full palmprint matching.

The generic model for palmprint identification systems is the same as illustrated in 1.14. It should be noted that the detailed architecture of a particular palmprint identification system depends on image resolution, image acquisition method, and features used. For example, sometimes it makes sense to apply noise removal in conjunction with ROI segmentation, and sometimes it does not. Similarly, whether palmprints need to be aligned before matching also depends on the exact application.

Regardless of image resolution, an efficient and cost-effective palmprint identification system takes five objectives into account, namely, financial cost, user acceptance, computational overhead, accuracy, and security. While low resolution methods are more friendly to users and require less computation, they lack in providing an adequate level of security. On the other hand, high resolution palmprints provide an advanced level of security and accuracy but at a great computational cost.

Whether low or high resolution, palmprints have quickly paved their way into mainstream biometric systems. The global biometric industry is expected to reach $100 Billion growing at a compound annual growth rate (CAGR) of 14.6% by 2030 [11]. Palmprints are a relatively less explored biometric but they are gaining importance rapidly on account of rich intrinsic features that are unique to a subject and permanent during a lifetime. Palmprint-based systems along with fingerprints were leaders in the biometric industry in the last decade and palmprint-based systems are expected to show a CAGR of 18.28% and touch around $2.6 Billion by 2030 [12]. As a result of the growing popularity of palmprints, the research community's interest in palmprints has also seen a sharp increase [9, 10, 13–15].

1.7 Palmprint Applications

Applications requiring low to middle-level security prefer the use of Low resolution palmprints. Easy and non-invasive image acquisition methods coupled with lower costs make them more acceptable to users. Application areas for low resolution palmprints include campus attendance systems, e-commerce, person registration, access control management systems, etc. But if a higher level of security is required, low resolution palmprints alone cannot suffice. In such cases, palmprints have to be fused with other biometrics to create a multi-modal authentication system that gathers information from multiple modalities [16][17]. Furthermore, low resolution palmprints can only be used in full-to-full palm matching. An illustration of a multi-modal biometric system is presented in Figure 1.15.

Applications requiring a higher level of accuracy and security prefer using high

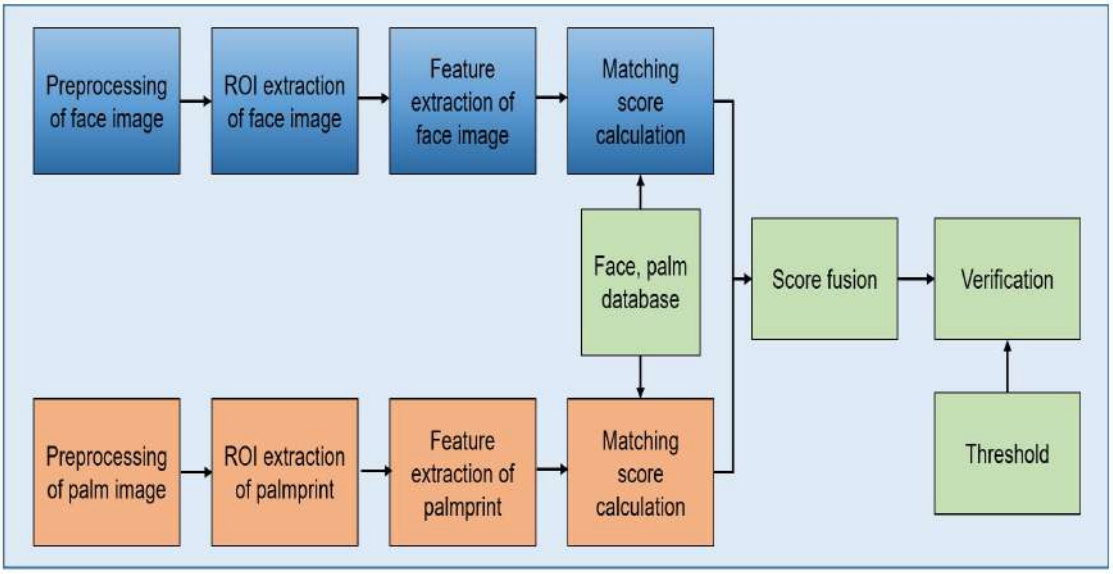

FIGURE 1.15: Multi-modal biometric system combining palmprint with face [18]

resolution palmprints. Minutiae-based level 2 features (Table 1.1) are more trustworthy and cannot be faked easily. High resolution palmprints can be divided into two main categories: full and latent. Full palmprints are usually acquired in a controlled environment using a scanner. Latent palmprints are found in crime scenes and have forensic value. They are usually extracted from weapons and vehicle steering wheels or windows etc. According to research conducted by the Federal Bureau of Investigation (FBI) in the USA, 30% of biometric evidence found in crime scenes is from palmprints [7]. Apart from an advanced level of accuracy, high resolution palmprints provide the additional advantage of latent-to-full matching. Whether full or latent, high resolution palmprints come with a lot of challenges. Even if they are acquired in a controlled environment, various factors like inconsistent flexibility of skin, and varying quality of ridge structure in different regions (Figure 1.12) make processing difficult. Figure 1.16 illustrates variations in ridge structure quality of within a palmprint.

Since high resolution palmprints are of large size, they have high computational overhead. The large size of the palmprint means more information to process. For example, high resolution palmprints provide approximately 8 times more minutia than a fingerprint [7]. Despite their challenges, high resolution palmprints provide the most reliable intrinsic features that have the potential to provide high identification accuracy in a uni-modal biometric system. Because of their high accuracy

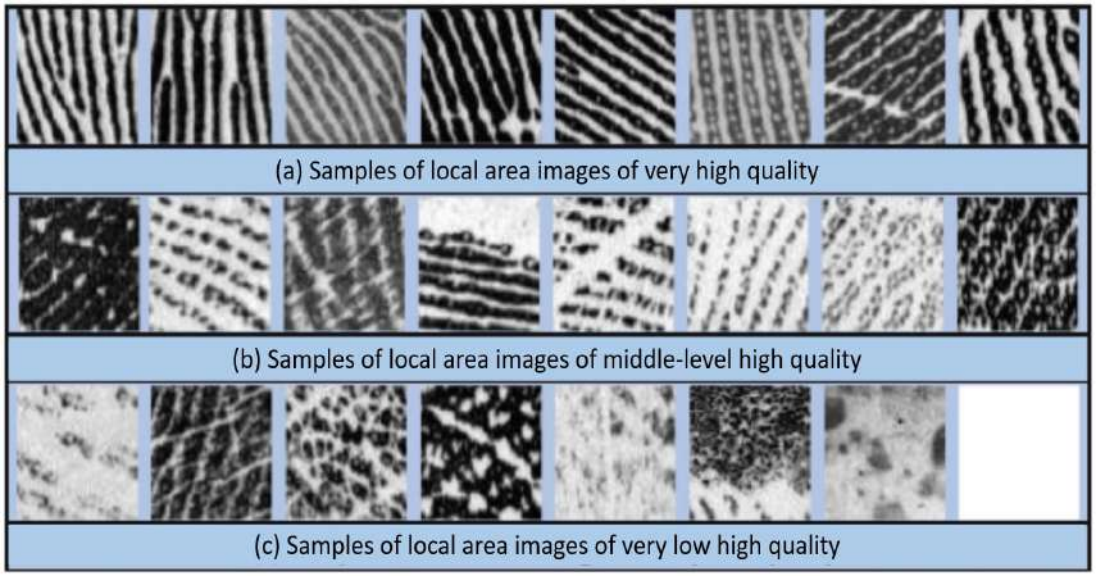

FIGURE 1.16: Different samples of local patches of palmprint depicting variation in ridge quality [15]

and forensic value, the focus of this thesis is on high resolution palmprints.

1.8 Palmprints vs Fingerprints

Components of a generic palmprint system are illustrated in Figure 1.14. First, the Region of Interest (ROI) is segmented from the image to limit subsequent processing to only the relevant areas of the image. Subsequently, image enhancement is performed which aims at accentuating identification features in the image. This is the most important step because error-prone enhancement deteriorates identification performance eventually. After enhancement, features are extracted and encoded and saved as templates in the database. During the online stage, features extracted from query images are compared with templates saved in the database, and a matching score is produced.

Since fingerprints and high resolution palmprints employ the same minutiae-based features for identification (Table 1.1), researchers have adopted algorithms used in fingerprint identification systems while working on high resolution palmprints. Fingerprint techniques used for ROI segmentation, enhancement, feature extraction, and matching have been tailored to work on palmprints. But palmprints differ from fingerprints in a few aspects:

- Large number of creases resulting in abrupt changes in ridge orientation
- Flexibility of skin resulting in abrupt changes in inter-ridge distance or ridge frequency
- Palmprints are much larger and contain much more information which increases computation complexity

Furthermore, high resolution palmprints provide a huge amount of information that encounters a lot of computation challenges. For example, a fingerprint image has a size of 512 x 512 in most datasets which provides 100 minutiae on average. On the other hand, a high resolution palmprint has a size of 2048 $\times$ 2048. On average, a palmprint would present 800 minutiae. This is exhaustive for the extraction and matching phases of any algorithm. There is another major issue of major and minor creases that makes high resolution palmprint enhancement very challenging. The presence of minor creases, especially in the thenar region (see Figure 1.12) affects the recovery of ridge structure adversely during the enhancement stage. Broken or poor ridge structure introduces spurious minutiae which not only increase computational requirements while feature extraction, encoding, and matching but also deteriorate identification accuracy.

In fingerprints, these creases either appear less frequently or are very thin. Ridge reconstruction or enhancement in fingerprints is much easier than in palmprints on account of width and number of minor creases. That is why fingerprint enhancement methods cannot be directly applied to palmprints. Jain and Feng [7] explained the detrimental effects of minor creases during the enhancement of palmprint images using a commercial fingerprint enhancement tool. One of their experimental results is shown in Figure 1.17. Verifinger is a commercially available software extensively used for ridge reconstruction in fingerprint images. While it performs excellently in fingerprints, it produces false ridges in palmprints. This further verifies the observation, that in palmprints the presence of large and small creases is a major issue that adversely affects the enhancement and feature extraction techniques. Hence, techniques used for palmprint enhancement should be tailored for palmprints.

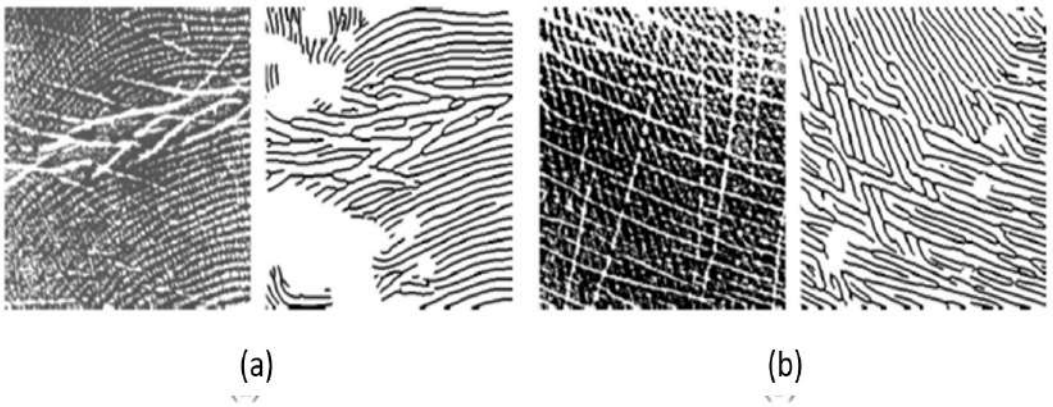

FIGURE 1.17: Creases in palmprints (a) a palmprint region with a major crease and its ridge skeleton created by Verifinger (b) a palmprint region with minor creases and its ridge skeleton created by Verfinger [7]

1.9 Purpose of the Thesis

The purpose of this thesis is to overcome the shortcomings of classical palmprint processing methods which were originally designed for fingerprints including ROI segmentation, enhancement, and minutiae processing that precede the matching stage. The motivation for the thesis stems from the idea that without effective ROI segmentation, enhancement, and selection of candidate minutiae, the accuracy of palmprint identification will deteriorate. Hence, this thesis proposes:

- An ROI segmentation method that effectively removes background pixels from the image to reduce processing and feature search space to only the palmar region of the image.
- An enhancement method that is adaptable to frequently occurring creases in the palmprint and varying ridge quality.
- A minutiae selection method to limit the number of minutiae that need to be matched to attain good matching accuracy.

1.10 Contributions of the Thesis

Major contributions of this thesis include:

- **Contribution 1:** A frequency domain palmprint Region of Interest (ROI) segmentation method as a pre-processing step to remove background noise from the palmprint. This reduces ROI to only the foreground pixels, thereby reducing subsequent computation.
- **Contribution 2:** A palmprint enhancement method that is inspired by recent work in image restoration using Convolutional Neural Networks (CNNs). It employs a two-step palmprint enhancement network *(PEN)* that is able to work on sufficiently large patches of palmprint, i.e., 96×96 without assuming the underlying ridge patterns to be stationary or uniform. In the first step, palm patches are classified by a CNN on the basis of dominant ridge orientation. In the second step, an image-to-image regression CNN converts patches directly into corresponding enhanced versions.
- **Contribution 3:** A novel matching algorithm that reduces the number of matches during the online matching stage resulting in reduced computation overhead. Since palmprints contain a lot of minutiae, running any matching algorithm on all minutiae can be cumbersome. This thesis proposes a minutiae selection algorithm that selects a limited number of minutiae for matching. So instead of matching all minutiae contained in a query and template image, a small subset of minutiae needs to be matched.

1.11 Structure of the Thesis

Chapter 2 reviews the literature available on low and high resolution palmprints. Since the focus of the thesis is on high resolution palmprints, the literature on low resolution is reviewed briefly while the literature on high resolution is reviewed in detail. The literature review encompasses all processes involved in a palmprint

identification system including Region of interest (ROI) segmentation, noise removal, enhancement, encoding, and matching.

In chapter 3, a frequency-based ROI segmentation method is introduced which sees the alternating ridge valley structure of the palm as a 2D sine wave and isolates dominant frequencies resulting in the removal of the background from the image.

Chapter 4 presents the proposed two-step enhancement method based on Convolutional Neural Network (CNN) in detail including material on the creation of training datasets for CNNs, training of CNNs during the offline stage, and online stages of CNNs presenting the overall flow of the proposed enhancement method.

Chapter 5 presents the results of the proposed method in two parts. In the first part, enhancement results are compared with state-of-the-art enhancement methods. The efficacy of the proposed method in recovering ridge structure even in high-noise areas (containing frequent minor or major creases) is illustrated. It is shown that while classical methods fail to maintain the continuity of ridges in noisy regions, the proposed method succeeds. The performance of the enhancement method in high ridge curvature areas is also illustrated. In the second part, the reliability of features (minutiae) extracted from the enhanced palmprint is established by evaluating matching accuracy. Chapter 5 also contains a discussion section on choices of architectures for CNNs used during enhancement.

Chapter 6 introduces the proposed minutiae selection algorithm that selects a small number of candidate minutiae for matching resulting in reduced computation during the matching stage. Chapter 7 concludes this thesis by listing the contributions of the proposed work and their efficacy as compared to state-of-the-art methods.

1.12 Summary

This chapter provides an overall introduction to this thesis. The importance of biometric technologies as a whole is presented along with the latest trends. Comparisons between popular biometric technologies are presented in terms of universality, user acceptance, cost, and collectibility. Later, palmprint is introduced as a

biometric technology with its historical use in human identification. The increasing utility and importance of palmprints as a biometric technology is argued because of their uniqueness and reliability. Challenges in designing palmprint solutions are discussed briefly. The next chapter provides an elaborate study of the literature on palmprints. After a rigorous literature survey research gap is identified and the objectives of the thesis are stated.

Chapter 2

Literature Review

There is abundant literature on the use of palmprints as biometric technology which can be divided into two main categories, i.e., High Resolution Palmprints and Low Resolution Palmprints. Due to a clear difference between high and low resolution palmprints, literature available on both types of palmprint forms two separate domains of knowledge. This is because physiological features, image acquisition sensors, and challenges presented by high and low resolution palmprints are totally different. The difference in features offered by palmprints at different image resolutions is listed in Table 1.1. Features available at low resolution are not available at high resolutions and vice versa. This is why methods pertaining to palmprint enhancement, encoding, and matching of high and low resolution palmprints are not comparable.

Although the focus of this thesis is on high resolution palmprints, this chapter briefly reviews the literature available on low resolution palmprints (contact-based and contact-less) as well for the sake of completeness.

2.1 Low Resolution Palmprints

Low resolution palmprint systems can be divided into two sub-categories, i.e., contact-based and contact-less systems. Contact-based palmprints are acquired in a controlled environment using scanners. Contactless palmprints are acquired

using commonplace cameras and enjoy increased user acceptability. Since they are acquired in an uncontrolled fashion, there are additional challenges due to rotation, translation, background illumination, etc. Below is a brief review of the literature available on both categories.

2.1.1 Contact-based Palmprint Systems

A typical contact-based palmprint is shown in Figure 2.1. Palms are scanned on specialized scanners that have pegs fixed in appropriate places. These pegs help in aligning the palms during registration and query. Aligning all palms simplifies subsequent enhancement and matching. In order to extract the palmar region

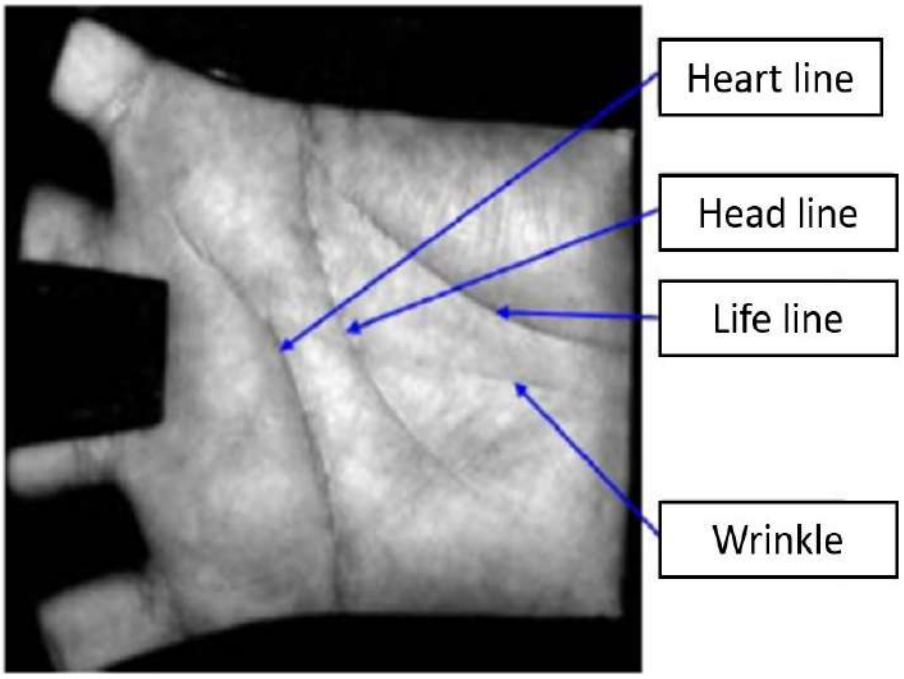

FIGURE 2.1: Contact Based Palmprint

from the image, almost every research work has used a Region of Interest (ROI) extraction method. This process restricts processing to only the palmar region of the image. This also helps to develop a common coordinate system for images for alignment before matching. The most popular method was proposed by D. Zhang et al [19]. Other methods are improvements of the same concept. The main idea in [19] is to find the boundary of the palm through a boundary-tracking algorithm. Then two reference points are calculated, which are the lowest points

in the arch formed between the index and middle finger and the ring and small finger, as depicted by red arrows in Figure 2.2 (c). A line segment joining these two reference points is drawn. Then a perpendicular bisector of this line segment is drawn to find the location of ROI as illustrated in Figure 2.2.

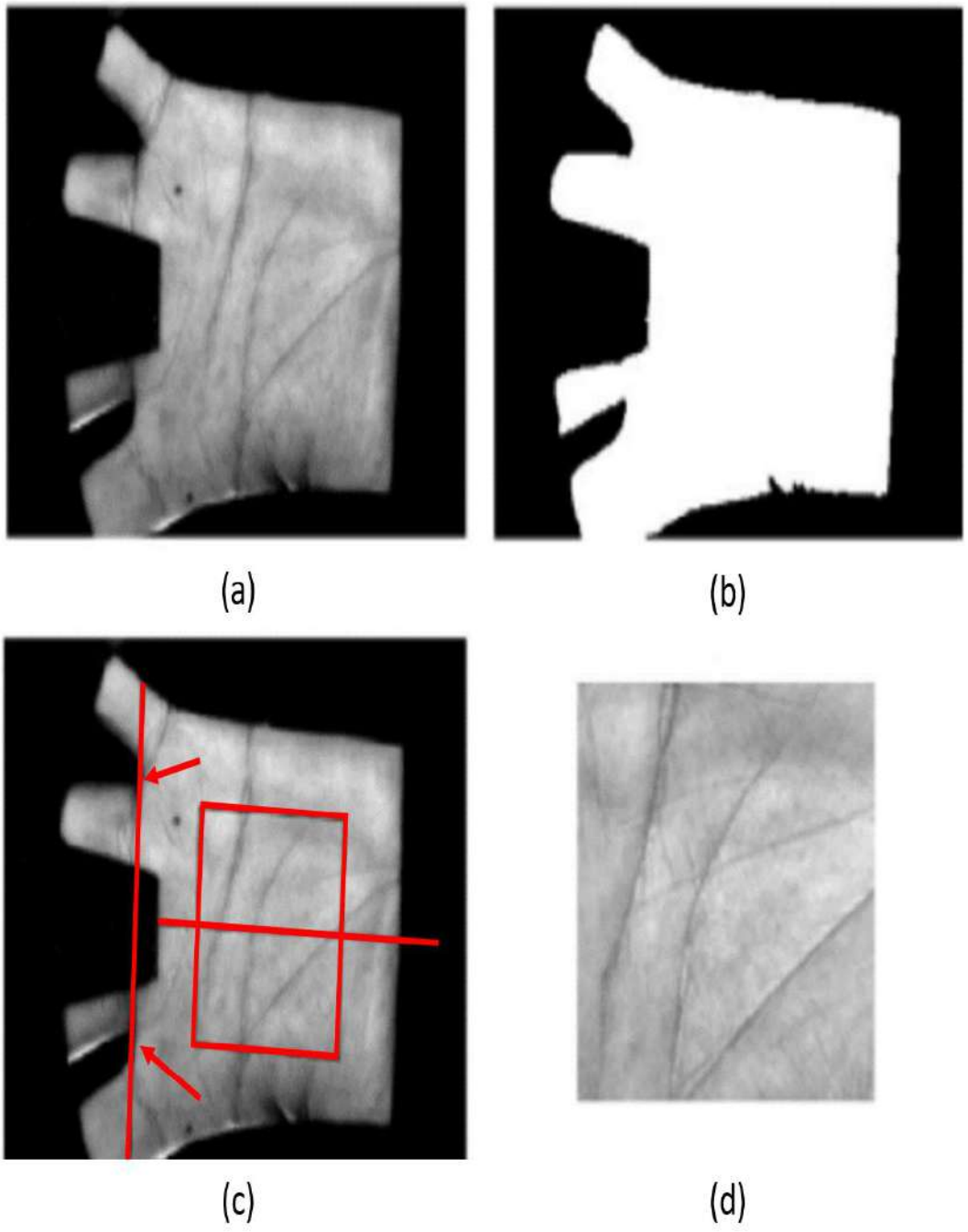

FIGURE 2.2: ROI segmentation: (a) Original Image, (b) binary image, (c) boundary tracking, (d) building a coordinate system, (e) extracting the central part as a sub-image, (f) resulting image [19]

Low resolution palmprints are texture-based images, and statistical characteristics

of these textures can become suitable features for identification. Many suitable local descriptors have been proposed by researchers which are mostly focused on finding and enhancing line features of the palm which are visible to the naked eye. The following features are most commonly used in low resolution palmprint systems:

- **Line Features**: Edge or line detectors are used to enhance and extract principal lines from the palm. Sobel, Difference of Gaussian (DoG), Radon, or Gabor filters have been commonly employed by researchers to enhance line features. The resulting image after convolution with these filters is a binary image. For calculating the similarity between a query and template image, hamming distance is used between the line features of both images. However, line features have not been found to be very accurate due to the fact that two subjects can have highly similar-looking principal lines.
- **Orientation-based Features**: Palmprint textures are full of features that have strong orientation characteristics. The basic assumption is that every patch of a palmprint contains a line (major or minor). When the orientation of the convolving filter (e.g. an oriented Gabor Filter) matches the orientation of the line present in the sub-image, the filter response is maximum. So, it is safe to infer that direction of the Gabor filter is the direction of the line. So lines in different regions of the palm are enhanced and their orientation characteristics are encoded and exploited for matching.

Gabor filters have been used to enhance line features extensively [20–22]. Radon filter has been used in [23] for the same purpose. Radon transform estimates the intensity of pixels around a line and depicts it as a peak. Hence, in the Radon transform, one sees peaks corresponding to lines in the original image. The location of these peaks in the transform gives the location of lines in the original image. Results of [23] are improved by the use of MFRAT (Modified Finite Radon Transform) in [24]. Difference of Gaussian (DoG) filters are used in [25]. A set of line detectors is prepared, based on the first and second derivatives of a Gaussian and convolved with the image. Figure 2.3 shows the results of line detection performed using DoG filters.

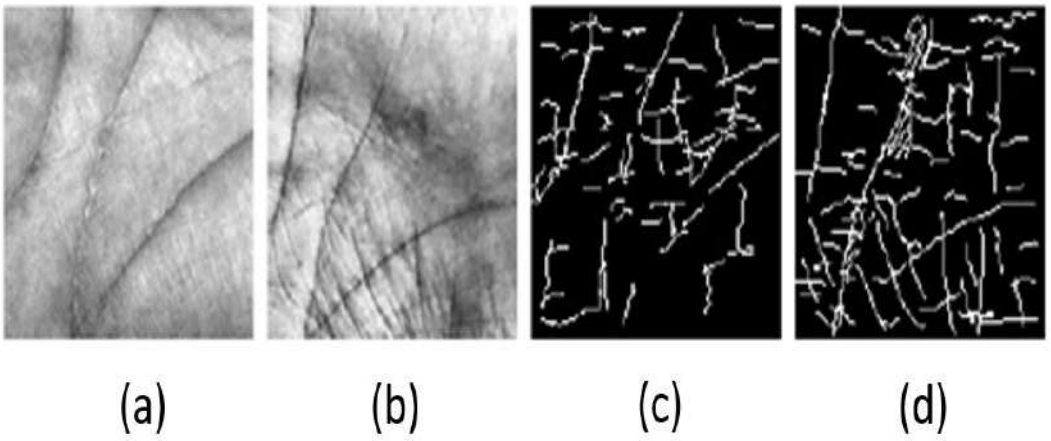

FIGURE 2.3: Results of [25] on PolyU dataset [26]. (a) and (b) are two palm images, (c) and (d) are corresponding edge images.

Figure 2.4 shows the results of applying MFRAT on images taken from [26]. MFRAT converts an image into its corresponding energy image. Lines can be detected based on the energy image. Wrinkles can be filtered out using a threshold on energy.

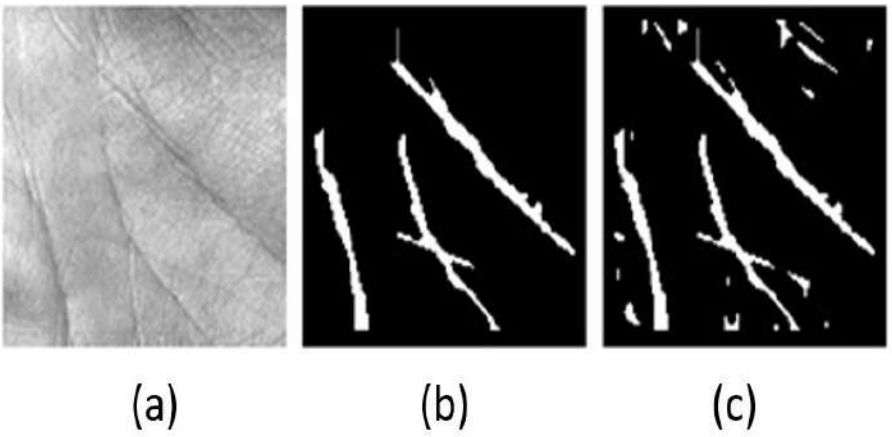

FIGURE 2.4: MFRAT [24] applied on PolyU dataset [26]. (a) Input Image, (b) Principal lines, (c) Wrinkles extracted by lowering the threshold.

Line enhancement algorithms are highly prone to noise and not very efficient in differentiating wrinkles from principal lines. To overcome this problem, researchers have exploited the orientation information present in the line features of the palm to improve identification results. Orientation information can be encoded with the position of features to create a palmprint template. The basic idea in orientation-based methods is to convolve the original image with a template that is oriented with an angle, say θ. After convolving, a mapping function converts the convolved

output in orientation descriptors into codes.

$$Orientation\_code(x, y) = fun[Conv\{T(\theta), I(x, y)\}] \tag{2.1}$$

Zhang et al. [19] used a single template 2D Gabor filter, with $\theta = \pi/4$ to extract a spectral orientation feature. Convolved features are then translated into binarized code using a mapping function. In [27], Kong and Zhang used 6 Gabor templates with different orientations instead of one. Gabor filter that matches the dominant information in the image gives maximum response or minimal convolved value. They called it competitive code (Figure2.5.

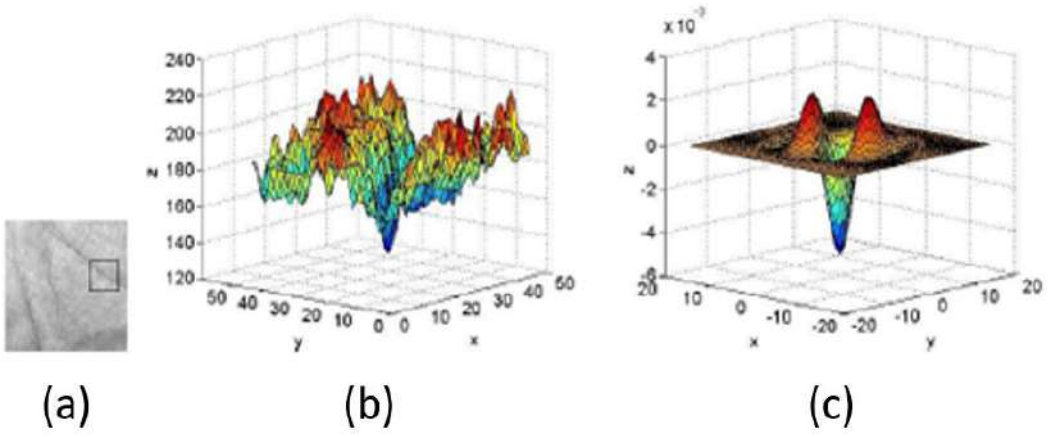

FIGURE 2.5: Theory of Competitive Code, (a) Sub image of palm, (b) Intensity value distribution of sub-image, (c) Gabor filter with similar orientation as the sub-image[27]

In practice, since orientations of Gabor filters are limited, i.e., in most cases, it is highly likely that no orientation will precisely define the orientation of the underlying line feature. Dominant line orientation in palm patches is usually related to the two closest Gabor filter orientations. Based on this observation, improvements were suggested by Fei et al. [28] who introduced Double Orientation Codes (DOCs). A lot of other improvements have been suggested by other researchers such as Guoet al. [29], Zhang et al. [30] and Fei et al. [31, 32]. Orientation-based methods translate dominant line orientation into binary codes. The hamming distance can be used to describe the similarity between them.

Recently, deep learning methods have also been employed in palmprint systems

[33] where local features in a small patch of palm are represented using features extracted from convolutional layers of deep CNN. Since features extracted from low resolution palmprints provide low to middle-level security, recent work on low resolution palmprints has been focused on improving the accuracy and security [9] of the system by fusing palm features with other physical features[10], thereby creating multi-modal biometric systems.

2.1.2 Contactless Palmprint Systems

Applications of contactless palmprints vary a great deal from contact-based palmprints. In contact-based methods, specialized scanners are used that have pegs fixed on them at appropriate places so that the palms of all subjects are aligned to a common coordinate system during enrollment and query (testing). In contactless palmprint systems, palms are not scanned, they are photographed using a variety of commercial devices (e.g. cameras, contactless scanners, etc.). Furthermore, images are taken in a free environment with no control over the rotation, translation, scale, and illumination of the image. This introduces intra-class variation which is not desirable. In order to improve the accuracy of contactless palmprint systems, additional features are incorporated. Figure 2.6 shows different types of contact-less palmprint images which illustrates the variation introduced during the acquisition of images.

As illustrated, due to an uncontrolled environment, effects like rotation, translations, illumination, scaling, etc. are introduced during image acquisition. Features thus chosen for matching should have a requisite level of robustness. Features generally used for matching in contactless palmprints are the following:

- **Scale Invariant Feature Transform (SIFT)**: SIFT [37] is an algorithm from the field of computer vision, originally developed for object identification. It has been used in various applications, for example, image stitching, gesture recognition, video tracking, etc. In the field of palmprint recognition, its ability to find local patterns in images is exploited. SIFT is totally invariant to rotation and scaling and is partially invariant to projection and

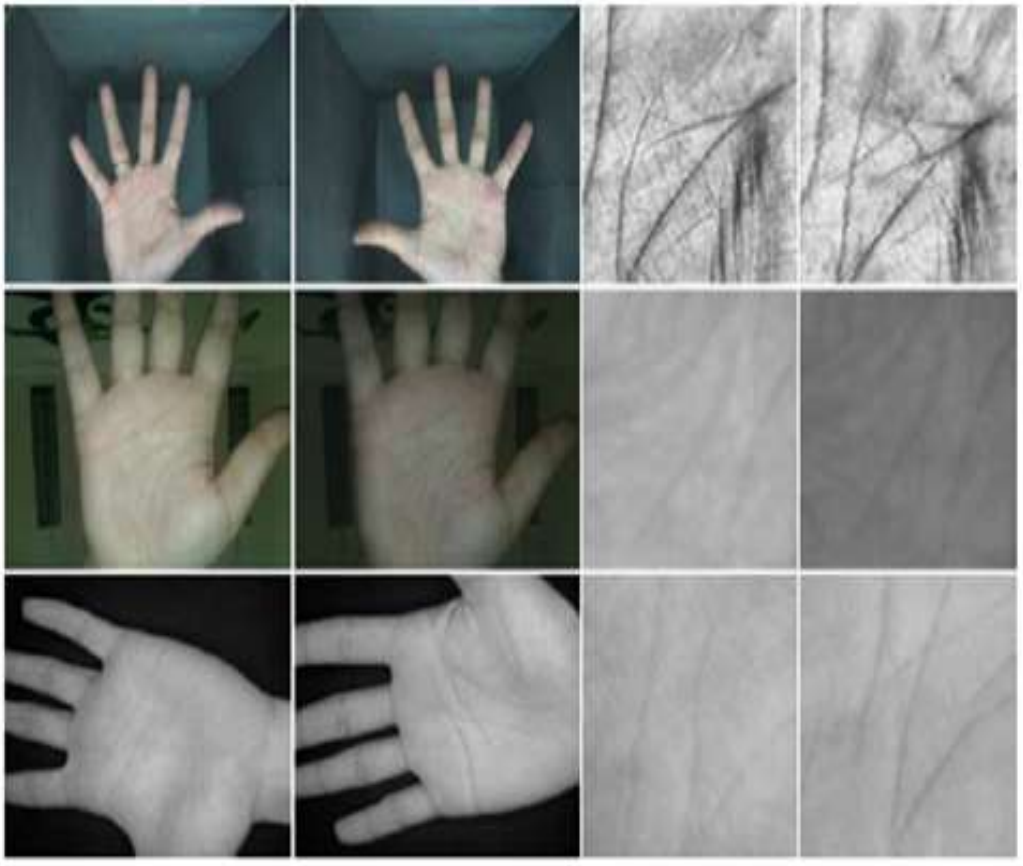

FIGURE 2.6: Contact-less palmprint images: First, second and third row are samples taken from IITD [34], GPDS [35] and CASIA [36] datasets. On the left are original palmprints and on the right are extracted Regions of Interest (ROIs).

illumination. SIFT aims to find stable key points across multiple scales of an image that have associated orientation information too.

- **Local Bit Pattern (LBP)**: LBP shows immense resilience to illumination variance along with total invariance to scaling and high robustness to rotation. The idea is to divide each image into sub-images. Pixel values of 8 neighbors are inspected around each pixel (center pixel). If the value of the neighboring pixel is greater than the center pixel, it is given a value of "1", otherwise "0". A histogram technique is used to define a texture and calculate a local binary pattern. LBP can be improved to make it rotation invariant as well.

- **Local Direction Pattern (LDP)**: LDP can be thought of as an improvised LBP adapted for images with rich line features (such as low resolution

palms). In LDP, a 3 x 3 neighborhood is convolved with 8 directional edge filters (Kirsch Edge Masks). Response to these filters is calculated and top k responses are selected. Based on the calculated response, an eight-bit code is defined which serves as a local texture descriptor. Bits corresponding to top k filters are set to 1 and the rest are set to 0.

Above mentioned features have been used extensively in contactless palmprint systems. For example, Wu et al. [38] exploited the robustness of SIFT towards variations in rotation and scaling. He first extracted the SIFT features and then discarded mismatched features between two palms. The final score in matching two palms is equal to the total number of matched SIFT features. Euclidean distance has been successfully used to match SIFT points in palmprints. A pair of points with descriptors p_i,q_j are taken as matched if:

$$d_{ij} < t_{min}(d_{ik}),\ k = 1, 2, ..., N,\ and\ k \neq j, \tag{2.2}$$

where d_{ij} and d_{ik} are the Euclidean distances between p_i,q_j and p_i,q_k and t is the threshold. Threshold t in the range $[0.58, 0.83]$ is suitable for contactless palmprint verification. Zhao et al. [39] improved the accuracy of SIFT descriptors. Based on SIFT points, he aligned two palmprints and devised a scheme to create competitive codes. The fusion of competitive codes with SIFT descriptors improves matching accuracy. Generally speaking, SIFT can be combined with other features as well for improved accuracy.

A directional gradient associated with LBP (DGLBP) was proposed by Michael et al. [40]. Four directional masks, i.e., vertical, horizontal, 45 degrees, and 135 degrees are convolved with the original image. Each of the resulting gradient images is divided into small sub-images and LBP is applied on each sub-window. The final descriptor is obtained using concatenated LBP of each sub-window of each gradient image. In [41] Lou et al. improved LDP to LLDP in which he used line extraction filters, namely, MFRAT and Gabor Filters at 12 different orientations. After convolution, encoding schemes like Enhanced LDP (ELDP)[42], Local Direction Number (LDN)[43] is used.

Results have shown that traditional methods for low resolution palmprint recognition do not perform as accurately on contactless palmprints as on contact-based methods. The reason is a high level of variations on account of translation, rotation, scaling, and illumination. Hence, traditional methods cannot extract the line features with the requisite amount of robustness. However, it is interesting to note that despite low accuracy of contactless palmprints, they have gained a lot of acceptability among users due to easy and non-invasive image collection methods.

2.2 High Resolution Palmprints

High resolution palmprints are the actual research area of this thesis. Most commercial applications of palmprint identification employ palmprints captured at 100 ppi or less. The use of high resolution palmprints for identification is a relatively less explored area. High resolution palmprints are taken at resolutions higher than or equal to 500 pixels per inch (ppi) thus making it possible to extract more detailed features of the palm resulting in lower identification errors. At this resolution, epidermal ridge lines of the palm are visible which are similar to fingerprints. Ridge structure in the palmprint has been the prime focus of research on high resolution palmprints. Ridge characteristics, namely, ridge orientation, ridge frequency, ridge endings, and ridge bifurcations are extracted after enhancing palm ridge structure. Ridge endings and bifurcations are called "minutiae" points. It is these minutiae that are used for palmprint identification. A typical high resolution palmprint is shown in Figure 2.7.

High resolution palmprints can be acquired using scanners in a controlled fashion or can be extracted from flat surfaces where they might leave traces. Accurate recovery of ridge structure in the palm is the most vital step and prime focus of this thesis. Subsequent minutiae extraction, encoding, and matching are contingent upon the success of ridge enhancement algorithms. All ridge enhancement algorithms available in the literature depend on one fundamental step: Estimation of local ridge orientation and frequency. Usually, both ridge orientation and frequency are estimated in a local fashion. After the estimation of local ridge

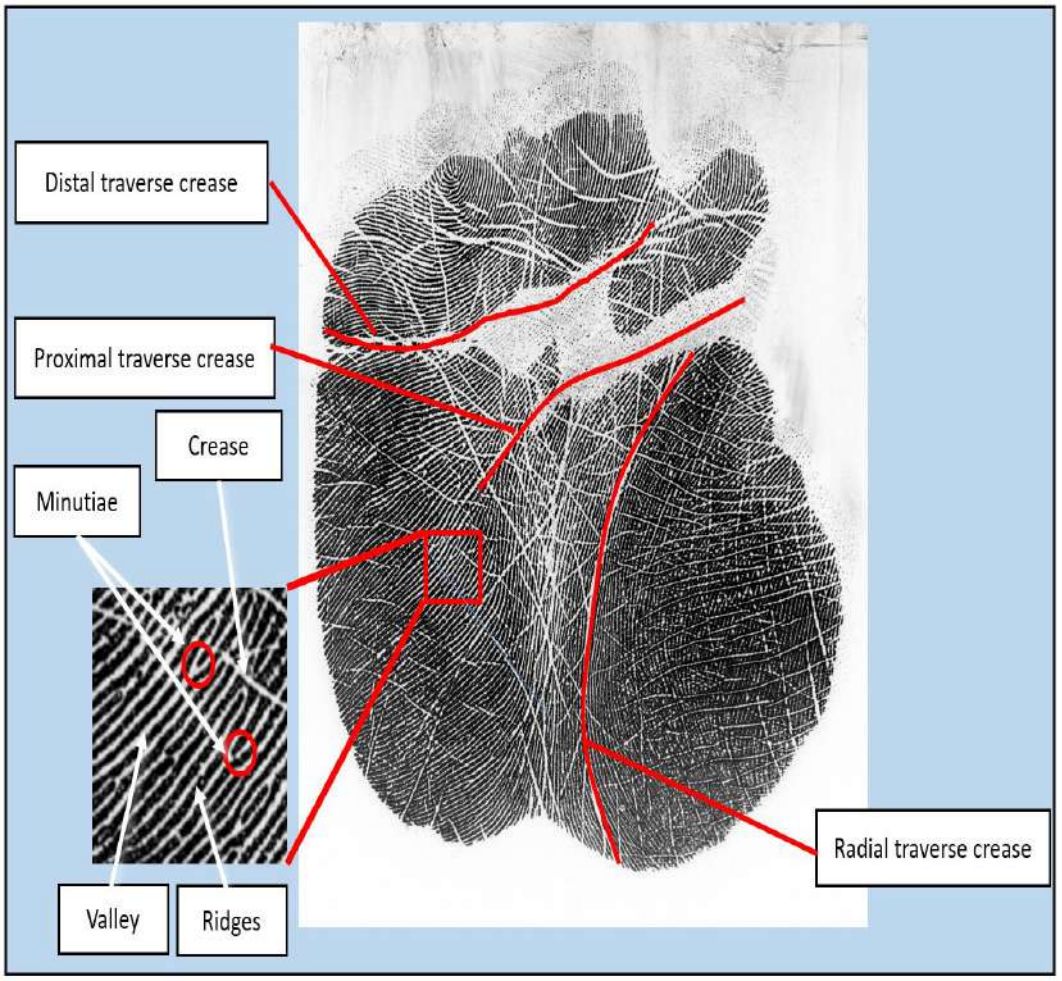

FIGURE 2.7: A high resolution palmprint [8]

orientation and frequency, appropriate contextual filters are applied to enhance ridge structure and subdue creases or any other artifacts that may be present. After enhancement of ridge structure, binarization and thinning of ridges is carried out. In the end, minutiae are extracted from thinned palmprint. The generic flow diagram of a biometric system presented in Figure 1.14 can be tailored for high resolution palmprints as illustrated in Figure 2.8.

Since features contained in high resolution palmprints bear a lot of resemblance with features contained in fingerprints (Level 2 features listed in Table 1.1), almost all steps shown in Figure 2.8 are borrowed from the field of fingerprint enhancement and identification and customized to work on palmprints. All of these steps form separate areas of research. Recent work on high resolution palmprints has focused on palmprint matching efficiency and limited novelty has been introduced in palmprint enhancement. The following sections of this chapter review major

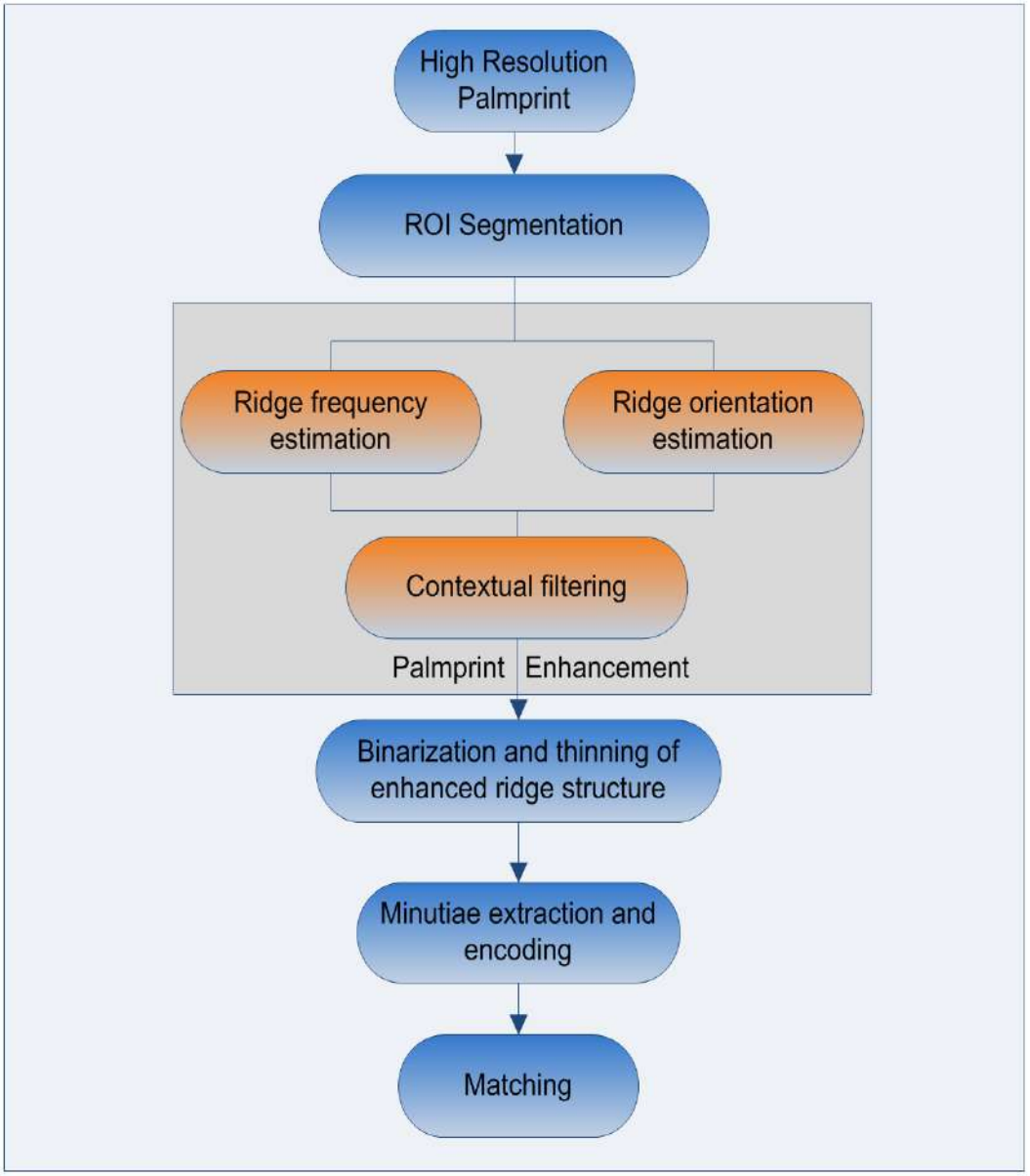

FIGURE 2.8: Generic processes of a high resolution palmprint system

work done in these research areas.

2.2.1 ROI Segmentation

The foremost step in any biometric system is the extraction of the Region of Interest (ROI) from the image. ROI is also called the foreground of the image. ROI segmentation serves various purposes: Firstly, it restricts processing to only the valid parts of the image resulting in reduced computation, secondly, it mitigates the possibility of mistaking textures or noise present in the background as intrinsic

features. Hence, ROI segmentation reduces the feature search space.

High resolution palmprints are of large size, usually, 2040×2040 [44]. This poses computational challenges at every step. It is therefore essential to extract ROI in the image before proceeding further. ROI in high resolution palmprint is the region containing palmar ridge lines. Palmprints extracted from real-world scenarios such as crime scenes suffer from multiple degradations. There are complex background textures interfering with the ridge structure of the palm. Segmentation of ROI not only reduces feature search space to the image portion containing palm ridge lines but also ensures extraction of valid features from the image instead of some artifacts that may have been introduced due to noise.

ROI segmentation techniques used in palmprints are borrowed from fingerprint techniques. Most popular methods exploit textural information available in the image to distinguish between foreground and background pixels. In [45], Maio and Maltoni proposed a local variance-based strategy for segmentation. Variance in image intensity within a patch of the image is calculated and compared against a threshold. If the variance is below the threshold, that area is classified as background otherwise it is classified as foreground. In order to speed up the process, non-overlapping patches are used. The variance of pixel or gray level intensity is given by the equation

$$Var(b) = \frac{1}{T^2} \sum_{i=0}^{T-1} \sum_{j=0}^{T-1} \left(G(i,j) - M(i,j)\right)^2 . \tag{2.3}$$

In equation 2.3, $Var(b)$ is the variance of a patch b, $G(i,j)$ is pixel intensity at (i,j), and $M(i,j)$ is the mean. The same concept is further extended by Bazen et al [46], where local variance is fused with the mean and coherence of ridge directions to find foreground pixels. Results are shown in Figure 2.9.

The same textural segmentation methods have been applied in high resolution palmprints [47, 48]. However, for palmprints, methods employing local thresholds prove computationally intensive due to the large image size. Results of grayscale variance-based methods on palmprints are presented in Figure 2.10

It can be seen in Figure 2.10, that some background pixels have also been mistaken as ROI. This is due to the fact that palmprints captured from real scenarios contain

FIGURE 2.9: Segmentation results of [46] on fingerprints

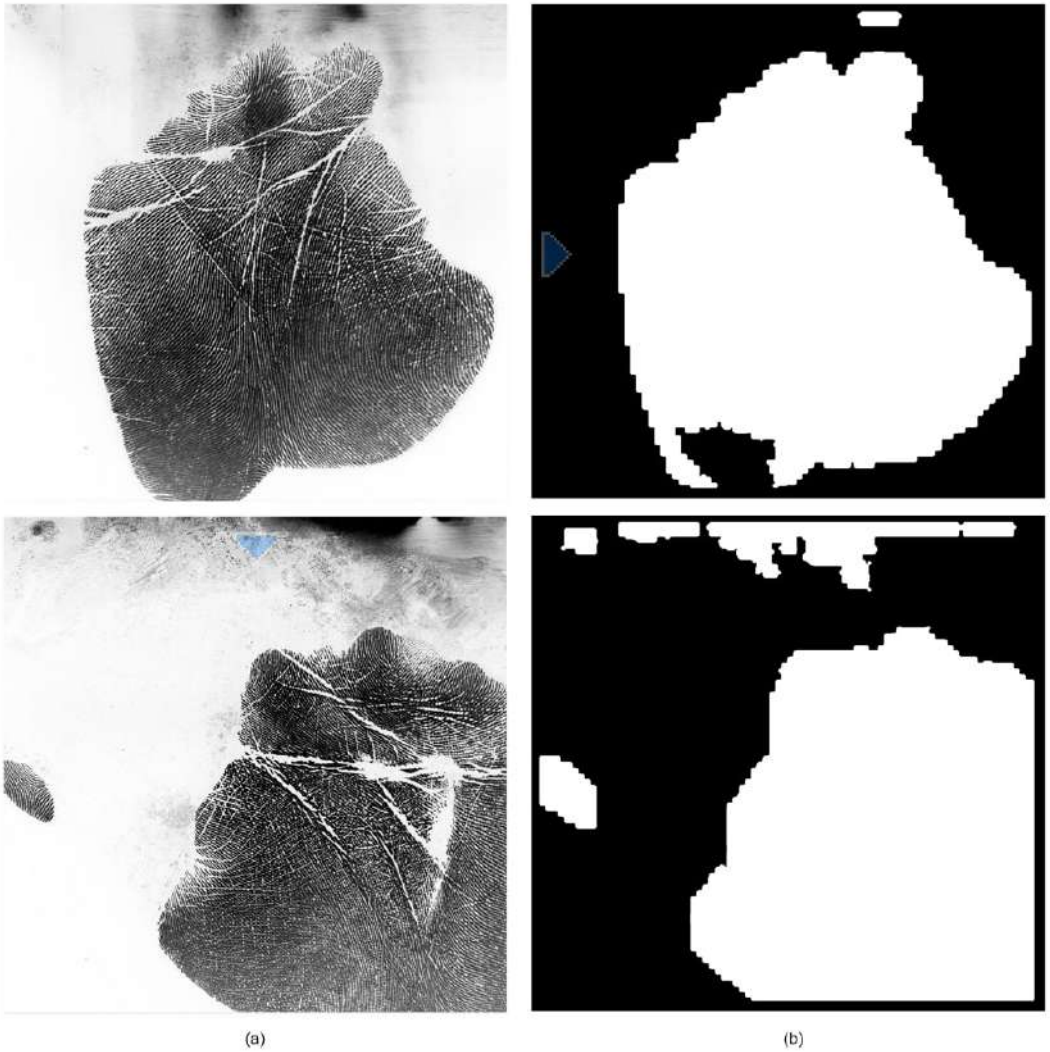

FIGURE 2.10: Segmentation results of [46] on palmprints: (a) Original Palmprint. (b) ROI mask

complex textures in the background. Sometimes, variance in grayscale levels of background pixels also matches the foreground pixels. As a result, the background

is mistaken as the foreground. In order to avoid this issue, a frequency-based ROI segmentation method is proposed in Chapter 3 which aims to identify image portions containing ridge lines rather than the variance of gray levels. Results show that the proposed method is able to extract a much sharper ROI mask which further reduces search space.

2.2.2 Initial Pre-processing

Like fingerprints, palmprints suffer from various degradations. One such degradation is low contrast between ridges and valleys. In order to improve contrast, sometimes segmented images are normalized. Normalization can be carried out globally or locally. This step standardizes gray levels in the image by giving desired mean and variance to image intensities. Normalization is sometimes referred to as contrast stretching. If M_o and V_o are the desired mean and variance respectively, normalization is expressed as:

$$\begin{aligned} I(i,j) &= M_o + \sqrt{\frac{V_o}{V}(g(i,j)-M)^2}, \;\; if g(i,j) < M, \\ I(i,j) &= M_o - \sqrt{\frac{V_o}{V}(g(i,j)-M)^2}, \;\; if g(i,j) > M. \end{aligned} \tag{2.4}$$

In equation 2.4, $g(i,j)$ is the actual gray level at (i,j) while $I(i,j)$ is the normalized gray level. Such steps improve contrast in the image which in turn makes ridge lines more pronounced. This helps in mitigating image degradations introduced by faulty image capture or non-uniform contact with scanners by users. But they do not cater for the degradations in ridge structure caused by creases, scars, or flexibility of the skin. To recover from these degradations in the ridge structure, dedicated enhancement methods are used which are the prime focus of this thesis. An example of mean-variance normalization is presented in Figure 2.11. Normalization of the images can be done during the enrollment stage. This

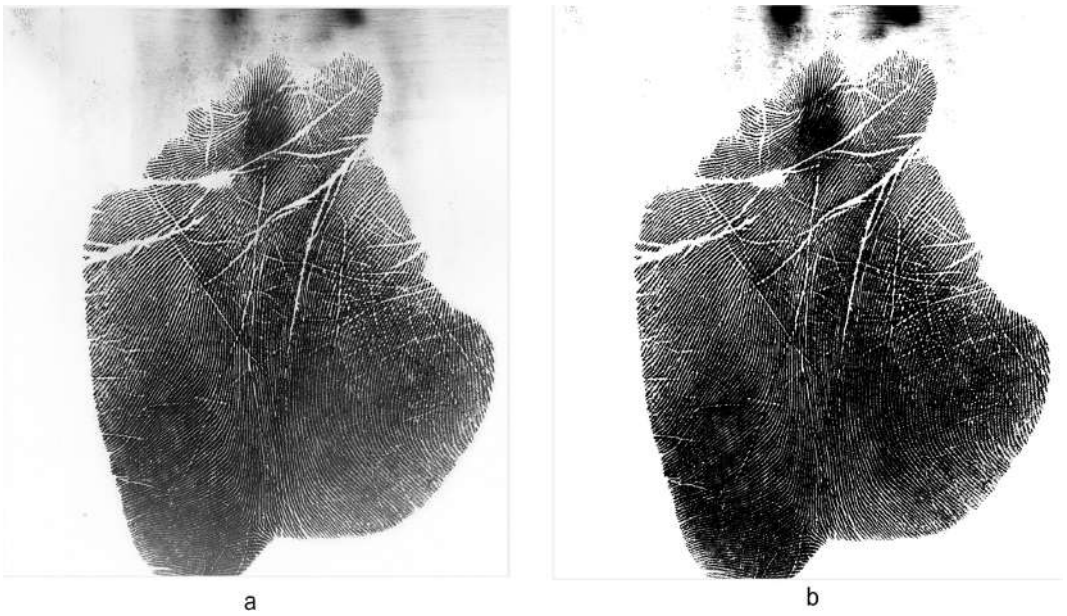

FIGURE 2.11: Results of normalization [49] on palmprint after ROI segmentation [46]

reduces computational requirements during the online matching stage. During the online stage, only query image needs to be normalized to bring a range of image intensities in conformity with the template images saved in the database.

2.2.3 Ridge Orientation Estimation

Ridge orientation estimation is the first step in palmprint enhancement (Figure 2.8). In high resolution palmprints, ridge lines are the most important feature. They are present everywhere in a palmprint and exhibit a continuous pattern of alternating ridge/ valley structure. An important property of the ridge lines is their orientation. Ridge orientation is not constant or stationary all across the palmprint. As can be seen in Figure 2.11, ridge orientation is different in different regions of palmprint. In regions of high curvature, ridge orientation changes abruptly. Figure 2.12 highlights abruptly changing ridge orientation. Since ridge orientation is not constant, algorithms designed for the estimation of local ridge orientation work in a local fashion, i.e., they adapt according to the underlying ridge structure in a local area. Subsequent filtering and final results of palmprint enhancement depend on the accuracy of ridge orientation estimation. That's why

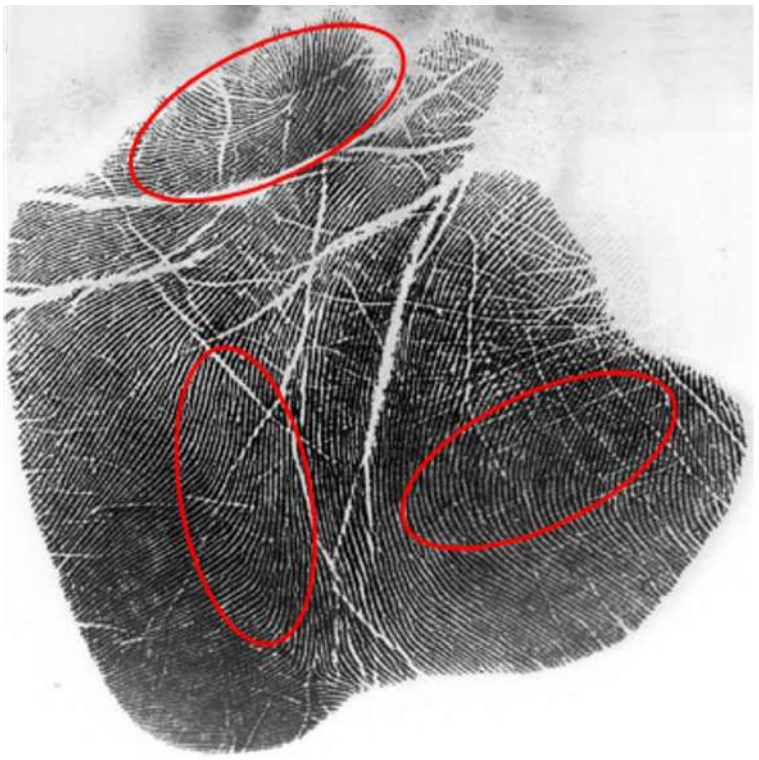

FIGURE 2.12: High curvature areas in a palmprint with abruptly changing ridge orientation

the estimation of local ridge orientation is at the core of the palmprint enhancement problem.

Gradient-based estimation of ridge orientation is by far the most popular method found in the literature. These methods were initially designed for fingerprints [50–53] which were later improvised for palmprints [47, 48, 54]. In the estimation of local ridge orientation, an orientation image or map $O(x, y)$ is created which contains values of ridge orientation at all values of (x, y). This gives dominant ridge orientation $\theta_{i,j}$ with respect to the x-axis at all pixels of the image as shown in Figure 2.13. The orientation image of map $O(x, y)$ thus created is used in subsequent contextual filtering, feature extraction, and feature encoding as well. This reiterates the importance of accuracy achieved in the estimation of ridge orientation as all subsequent steps involved in a palmprint system depend on it.

In gradient-based methods, ridge orientation estimation is done in the spatial domain [47, 48, 51, 54, 55]. Palmprint is divided into small patches, e.g. 17×17 centered at (x_i, y_j), and x and y gradients are computed. Horizontal and vertical gradient images are expressed as equation 2.5 and 2.6 respectively, i.e.,

$$G_x = \sum_{h=-8}^{h=8} \sum_{k=-8}^{k=8} \nabla_x (x+h, y+k)^2, \quad and \tag{2.5}$$

$$G_y = \sum_{h=-8}^{h=8} \sum_{k=-8}^{k=8} \nabla_y (x+h, y+k)^2. \tag{2.6}$$

Where ∇_x and ∇_y are x and y gradient components of the palmprint image computed using a Gaussian filter. ∇_x and ∇_y give the magnitude of intensity change at each (x, y). G_{xy} is the product of gradients given by equation 2.7. Product of G_x and G_y is calculated as

$$G_{xy} = \sum_{h=-8}^{h=8} \sum_{k=-8}^{k=8} \nabla_x (x+h, y+k)^2 \times \nabla_y (x+h, y+k)^2. \tag{2.7}$$

Other than magnitude, gradients have orientation properties too. The direction of the gradient at a given point gives the direction of maximum change in pixel intensity. So, after calculating gradient magnitude values at each point in the image, the direction of the gradient is also calculated. Based on these, an orientation map $O(x, y)$ is created. $O(x, y)$ is constructed by taking the inverse tangent of the gradients of the palmprint image as given by the equation

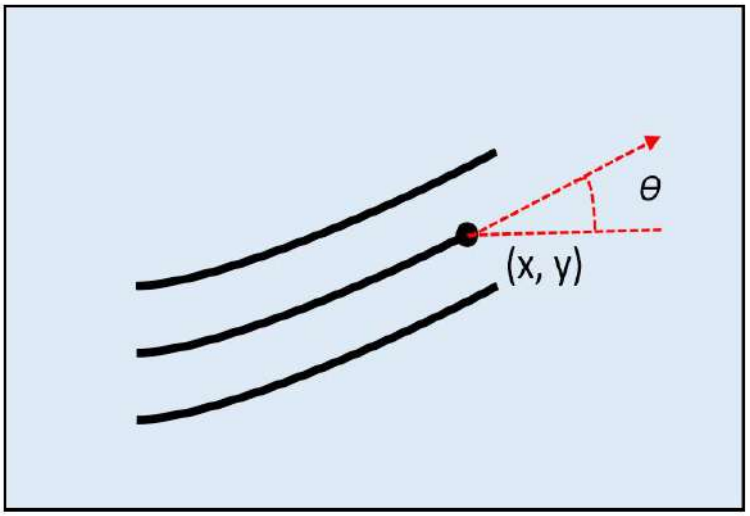

FIGURE 2.13: Ridge orientation at a point[50]

$$O_{xy} = 90^{o} + \frac{1}{2} tan^{-1}\left(\frac{2G_{xy}}{G_x - G_y}\right). \tag{2.8}$$

Gradient-based methods work fine for fingerprints but a large number of creases present in the palmprint deteriorate their performance. This is because gradient calculation in palm regions with frequent creases picks up directions of creases rather than ridges and as a result a faulty orientation map $O(x, y)$ is created. An example of orientation estimation is presented in Figure 2.14.

An alternative method for finding ridge orientation is the region-growing method. Unlike gradient-based method, region-growing method is a frequency domain method. Local ridge pattern can be modeled as a 2D sine wave [7]. Fourier analysis of a

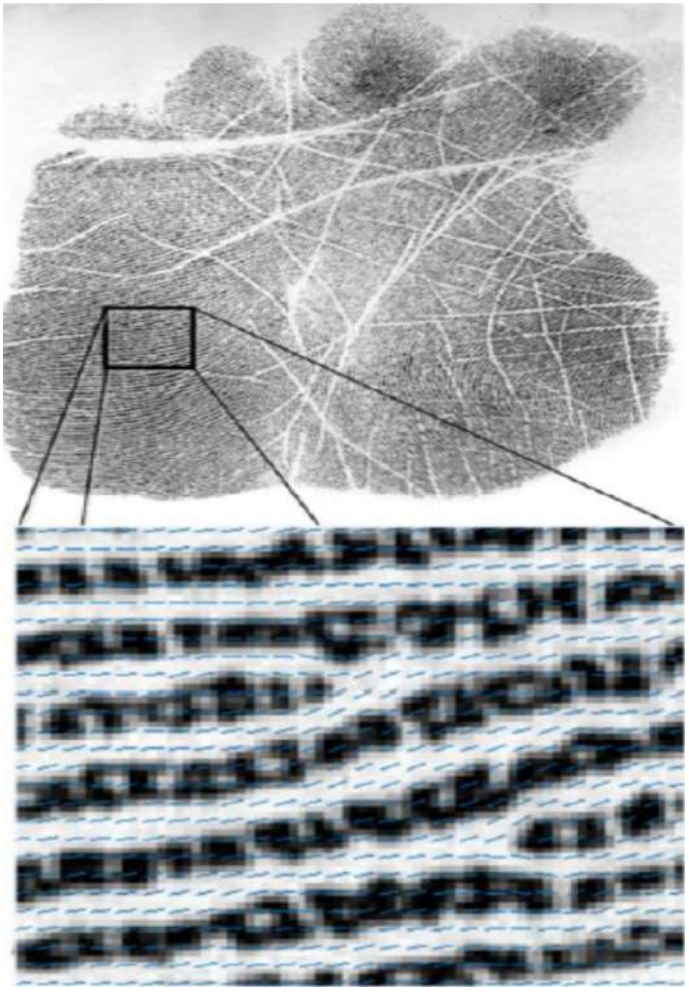

FIGURE 2.14: Ridge orientation estimation depicted in a region of palmprint by blue lines [48]

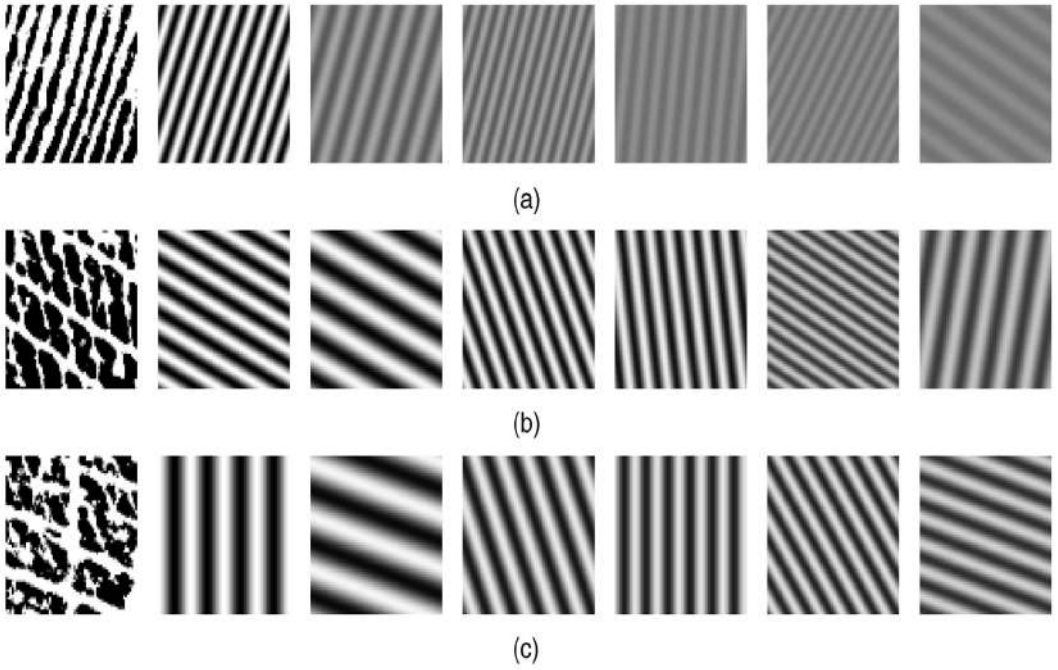

FIGURE 2.15: 6 strongest sine waves corresponding to variable quality patches(a) good patch with no crease (b) creases in one direction (c) creases in two directions (d) thinned patch with minutiae [7]

2D sine wave shows peaks corresponding to the frequency and orientation of the sine wave. Using this property of the Fourier transform, the local ridge orientation and frequency of a palm patch can be estimated simultaneously. Different sizes of patches have been used by different researchers. For example, Jain et al. [7] used an 8×8 patch while Dai et al. [56] used a patch size of 64×64. Variations of the same method have been used by [57, 58]. Patches with clear ridge structures produce clear peaks in the frequency spectrum showing the orientation of the underlying ridge lines. But patches with creases show multiple peaks in the frequency spectrum. In patches with creases, it is difficult to ascertain which peak corresponds to the underlying ridge pattern. An example of modeling ridge lines as 2D sine waves is presented in Figure 2.15.

It can be seen in 2.15, that in patches with creases, strongest sine waves correspond to creases and not the ridges. These sine waves are produced from the peaks observed in frequency spectrum of the patch. Good quality patches are classified as ''seeds" and they are joined with adjoining patches using a region growing algorithm to produce a complete orientation map $O(x, y)$. An example of orientation map created using region growing method is presented in Figure 2.16. However,

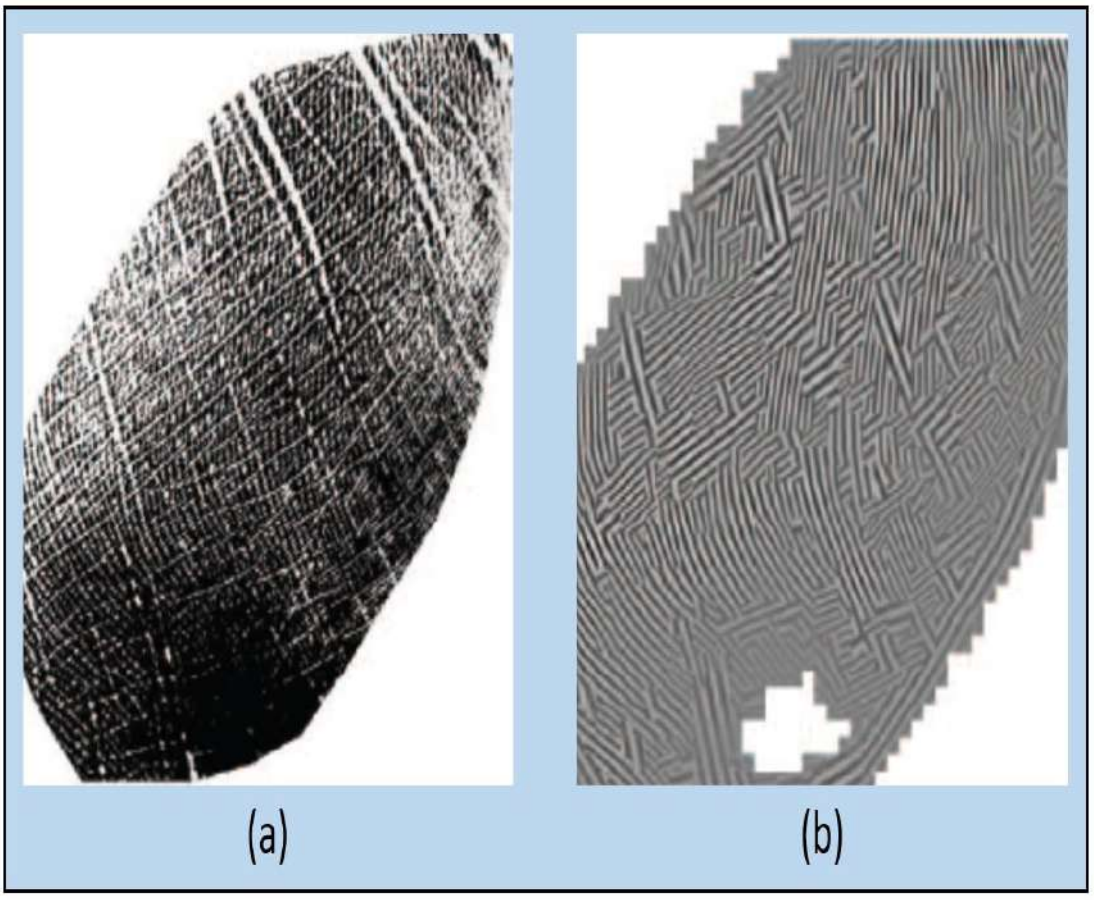

FIGURE 2.16: Orientation map of a partial palmprint made from 1st strongest sine wave of patches 64×64 [7]

these methods assume underlying ridge pattern to be stationary or constant which is a risky assumption in case of palmprint. Large number of creases and flexibility of skin can introduce abrupt changes in ridge pattern even in a patch as small as 64×64.

Another less popular frequency domain method of estimating local ridge orientation is through Gabor filter banks [59]. This method shares the same basic idea as the region-growing method. The frequency spectrum of a small patch of palmprint is multiplied with a series of Gabor filters and the direction of the Gabor filter that gives the maximum is considered as the direction of the underlying local ridge pattern. However, this method is costly in terms of time and computation. Whether orientation is created using gradient-based methods or region-growing methods, its accuracy is of prime importance for subsequent enhancement to succeed as orientation of local contextual filters used in enhancement is derived from the estimated orientation map $O(x, y)$.

2.2.4 Ridge Frequency Estimation

Ridge frequency estimation is the second step in palmprint enhancement (Figure 2.8). Ridge frequency or inter-ridge distance is also an intrinsic feature of the palm ridge structure. Ridge frequency estimation is as important as ridge orientation estimation for the enhancement of palmprint. While ridge orientation helps determine the orientation of the contextual filter used for palmprint enhancement, ridge frequency estimation helps determine width of the contextual filter. Just like ridge orientation, it is estimated in a local fashion and a ridge frequency map is estimated containing ridge frequencies of different neighborhoods of palmprint.

Since palms have flexible skin, ridge frequency is not constant throughout the palm. In general, it varies from 9 to 11 pixels in 500 ppi image [60]. Estimation methods for ridge frequency are also borrowed from fingerprints. Hong et al. [50] divided a fingerprint into small patches and estimated ridge frequency in the spatial domain by the x-signature of ridges within an oriented window as shown in Figure 2.17.

In this method, a window of a specified size is aligned with ridge orientation and the number of ridges is counted within the window. The total number of ridges is divided by the total distance (pixels) between the first and last ridge in the window. Let I be the palmprint image and O be the orientation map of the image. x-signature method of frequency estimation is explained below:

- Divide image I into small patches or blocks (say 16×16)
- Place a window of shape $l \times w$ at the center of a block centered at (i, j). Orientation of the window should be aligned with local ridge orientation identified from orientation map O (as shown in Figure 2.17).
- Calculate x-signature, $X[0], X[1], ..., X[l-1]$ for ridges and valleys contained in each block centered at (i, j), i.e.,

$$X[k] = \frac{1}{w} \sum_{d=0}^{w-1} I(u, v), \tag{2.9}$$

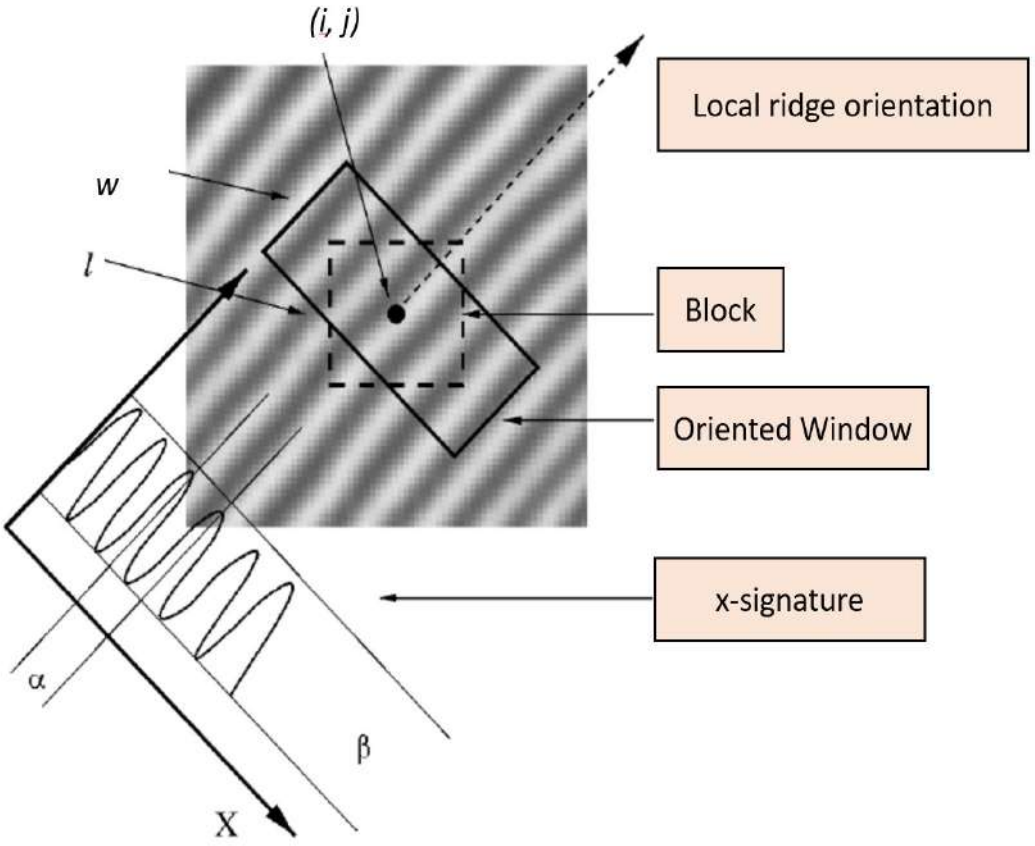

FIGURE 2.17: Ridge frequency estimation: oriented window and x-signature [50]

where

$$u = i + \left(d - \frac{w}{2}\right) \cos O(i,j) + \left(k - \frac{l}{2}\right) \sin O(i,j), \tag{2.10}$$

and

$$v = j + \left(d - \frac{w}{2}\right) \sin O(i,j) + \left(\frac{l}{2} - k\right) \cos O(i,j). \tag{2.11}$$

- For a block centered at (i, j) consisting of uniform ridge structure, x-signature takes the form of a discrete sinusoidal wave which has the same frequency as that of palm ridges. Then the frequency of ridges in the block centered at (i, j) is calculated by counting the number of pixels between two consecutive peaks in the x-signature. That is if $T(i, j)$ is the average number of pixels

between two peaks in the x-signature, the ridge frequency in the block is

$$F(i,j) = \frac{1}{T(i,j)}. \tag{2.12}$$

If peaks cannot be identified clearly in a block's x-signature due to a non-uniform ridge structure, the block is considered invalid. Non-uniform ridge structures can appear due to degradations in gray scales or occlusions in the palmprint or the presence of a large number of creases. In such cases, the frequency of the block is calculated using interpolation between neighboring blocks.

This method is straightforward and effective, but many improvements have been suggested to cater to fingerprints with scars or creases. For example, in [52] Chikkerur proposed a Short Term Fourier Transform (STFT) based analysis for simultaneous estimation of frequency and orientation in fingerprints. Later, ridge frequency normalization was also suggested by Ghafoor et al. [53] to cater for variations in frequency so that width of contextual filters can be standardized for the complete fingerprint. Apart from scars, creases, and occlusions, another reason for non-uniform ridge frequency is the flexibility of the skin. During acquisition, non-uniform application of pressure on the sensing area can result in non-linear distortion of ridge frequency. This problem is even more pronounced in the case of palmprints as palm skin is many a time more flexible than finger skin. Frequency normalization becomes useful in this scenario.

Process of frequency normalization proposed by Ghafoor et al. [53] is illustrated in Figure 2.18. Image is divided in to small blocks of $m \times m$ and filtered in frequency domain using directional filters whose direction is guided by the orientation image. This helps in the isolation of dominant ridge frequencies in the frequency spectrum. Once the frequencies corresponding to underlying ridge pattern have been estimated, each block is brought back into spatial domain and is scaled to bring ridge frequency as close to mean ridge frequency f_m as possible. The effect of frequency normalization on subsequent ridge enhancement is shown in Figure 2.19.

Techniques of ridge frequency estimation mentioned above were originally proposed for fingerprints and later improvised for palmprints. An original work on

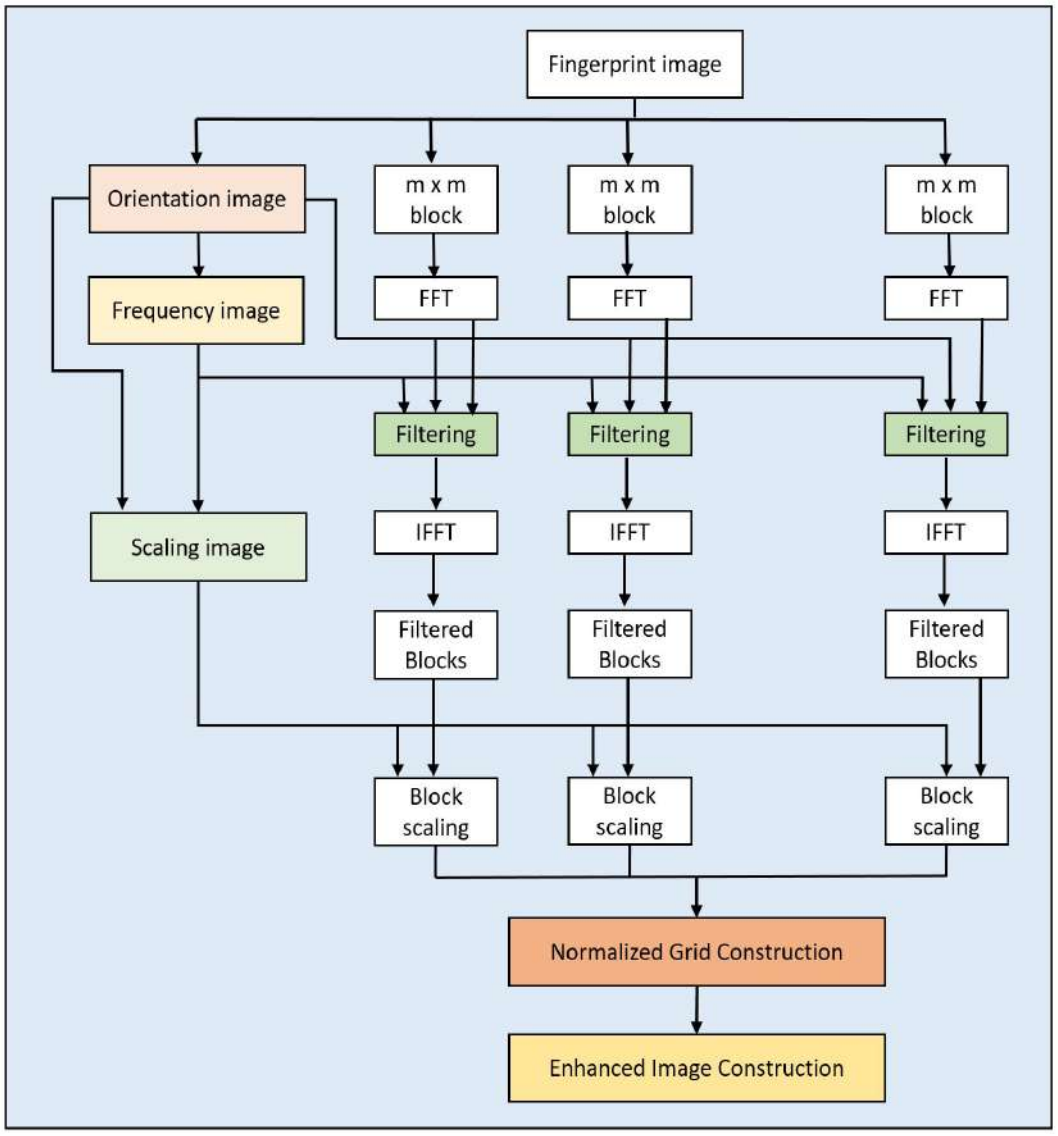

FIGURE 2.18: Ridge frequency normalization architecture [53]

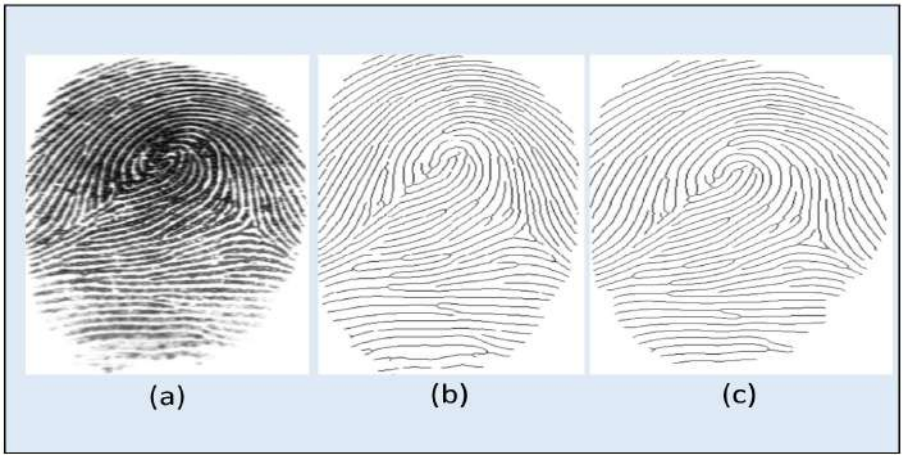

FIGURE 2.19: Ridge frequency normalization: (a) input fingerprint, (b), fingerprint enhanced without frequency normalization, (c) fingerprint enhanced after ridge frequency normalization [53]

palmprint ridge orientation and frequency estimation is the region growing method [7, 56–58] as already described in section 2.2.3 which estimates ridge orientation and frequency simultaneously by modeling ridge pattern in a block as a 2D sine wave. Orientation of the peaks pertaining to ridge pattern gives the orientation of the ridges while their distance from the center of DFT gives the magnitude of the ridge frequency.

2.2.5 Contextual Filtering

After the estimation of ridge orientation and frequency, the last step in enhancement is contextual filtering. As stated earlier, the direction and width of contextual filters are guided by ridge orientation and frequency estimates. It is imperative for filtering methods to be efficient in recovering genuine ridge structures in order to extract reliable features that improve identification accuracy. Error-prone filtering methods produce false features (minutiae) that deteriorate performance in identification and increase computational overhead. Whether gradient-based methods are used for the estimation of ridge orientation and frequency or region growing methods, recovery of ridge structure in palmprints or fingerprints is done using contextual filters. The most popularly used contextual filters are Gabor filters, cosine filters, or any other directional filters.

Most of the recent work in high resolution palmprints is focused on palmprint matching algorithms and limited novelty has been introduced in enhancement methods. As a result, most works still employ classical fingerprint enhancement methods on palmprints. After estimation of ridge orientation and frequency, Gabor filters are used in [7, 47, 48, 56, 57]. 2D Gabor filters are configured locally usually in a window of 17×17 pixels according to orientation and frequency of the underlying ridge structure. Gabor filter is in fact a Gaussian kernel modulated by a sine wave oriented at an angle. Gabor filters are expressed by:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) exp\left[jw_x(xcos\theta + ysin\theta)\right], \tag{2.13}$$

where w_x is the frequency of sinusoidal wave, θ is the orientation and σ is spread of Gaussian in both x and y directions. θ and σ depend on the orientation and frequency maps of the image. Figure 2.20 illustrates the shape of a 2D Gabor filter.

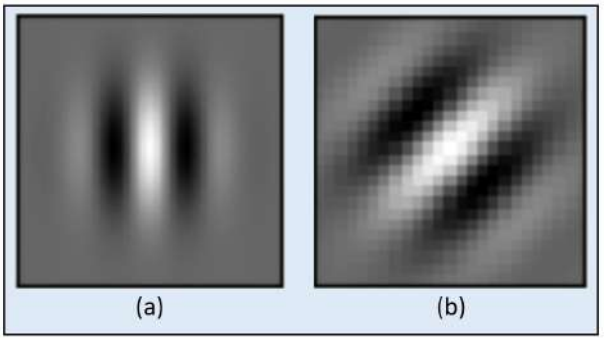

FIGURE 2.20: 2D Gabor filter: (a) $\theta = 0$, (b) $\theta = 45$

Directional filters can also be applied in the frequency domain [51]. The image is converted into its DFT equivalent and multiplied by a directional filter. The filter is applied in a local fashion where the orientation of the filter is changed according to the underlying ridge orientation which is derived from the orientation map. Figure 2.21 presents the results of contextual filtering on palmprints. Contextual filters are applied in small blocks. Each contextual filter is configured according to the orientation and frequency of the underlying ridge structure. A summary table of enhancement methods found in literature is presented in Table 2.1.

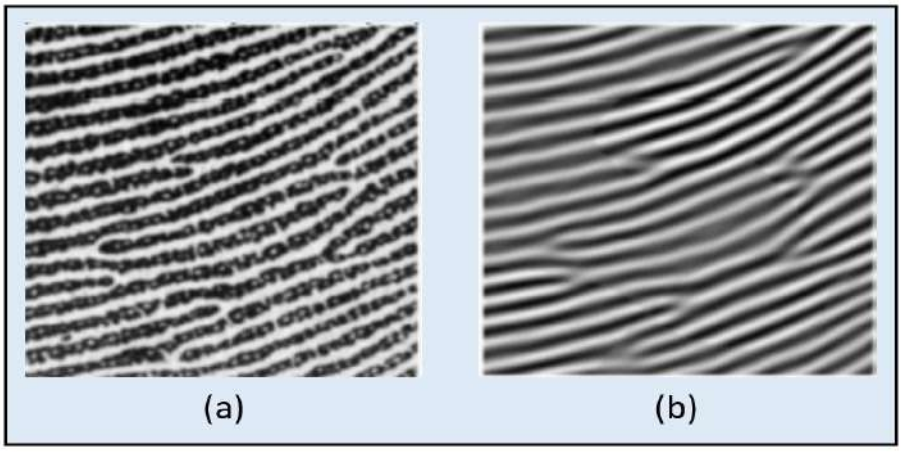

FIGURE 2.21: Contextual filtering:(a) palmprint patch, (b) Gabor enhancement [48]

Domain	Method	Authors	Orientation Estimation	Frequency Estimation	Contextual Filters
Spatial	Gradient-based	Hong et al.[50]	Pixel-wise gradients	x-signature	Gabor
		Maltoni et al.[55]	Pixel-wise gradients	x-signature	Gabor
		Cappelli et al.[54]	Pixel-wise gradients	x-signature	Gabor
		Ghafoor et al.[51]	Patch-wise gradients	x-signature	Directional filters plus Gabor
		Tariq et al.[47]	Patch-wise gradients	modified x-signature	Gabor
		Ghafoor et al.[48]	Patch-wise gradients	modified x-signature	Gabor
		Ghafoor et al.[53]	Patch-wise gradients	modified x-signature	Gabor
Frequency	Region-growing	Jain et al. [7]	DFT Analysis	DFT Analysis	Gabor
		Dai et al. [56]	DFT Analysis	DFT Analysis	Gabor
		Dai et al. [61]	DFT Analysis	DFT Analysis	Gabor
		Soleimani [57]	DFT Analysis	DFT Analysis	Gabor
		Hussein et al. [58]	DFT Analysis	DFT Analysis	Gabor
	STFT	Chikkerur et al. [52]	STFT Analysis	STFT Analysis	Raised cosine
	Gabor Banks	Jain et al. [59]	Gabor bank	Gabor bank	Gabor

TABLE 2.1: Summary of enhancement techniques

2.2.6 Binarization and Thinning

Sections 2.2.3, 2.2.4 and 2.2.5 constitute the enhancement phase of palmprint systems. Once palmprints are enhanced, the next phase is to extract features that can be encoded and used in the identification process. The first step in this phase is binarization and thinning (Figure 2.8). Binarization is the process where a multi-level grayscale image is converted into a black-and-white image. After conversion, all ridges are represented by white pixels whereas valleys are represented by black pixels. The decision of classifying pixels as black or white is done on the basis of a threshold [55, 62]. Selection of thresholds is made easy by filters used for enhancement as they reduce illumination variation in the image and produce an output image with zero mean. So zero becomes a suitable threshold for binarization. Pixels with intensities above that threshold are classified as white while pixels with intensities below that threshold are classified as black. Figure 2.22 illustrates how an enhanced palmprint is binarized.

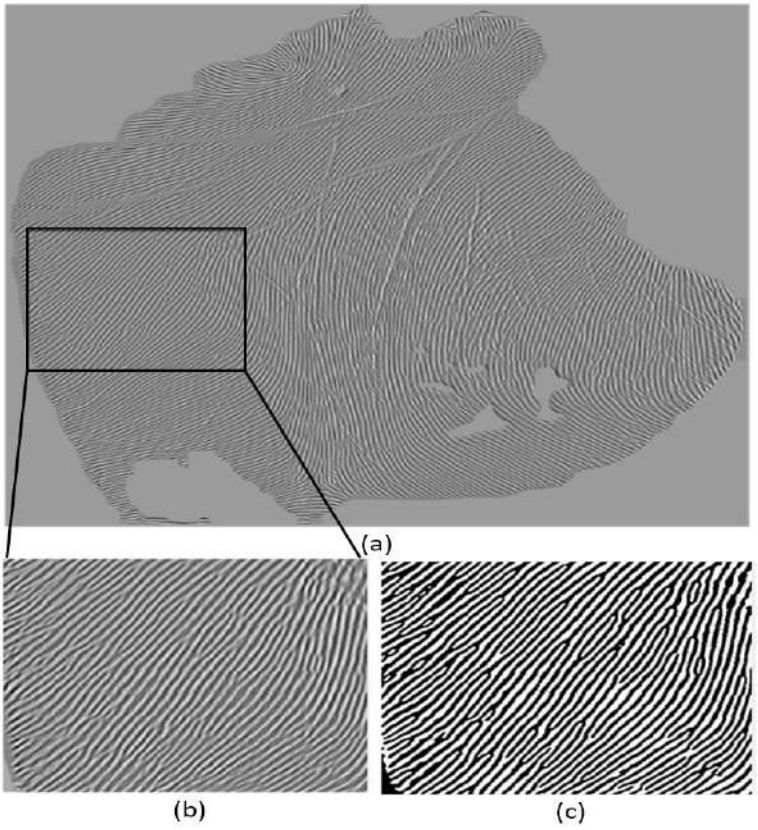

FIGURE 2.22: Palmprint binarization: (a) enhanced patch (b) binarized patch

The final step before feature extraction is thinning of the binary image. Thinning

is the process of reducing ridge width to just 1 pixel. This forms a skeleton image that keeps ridge continuity intact. This thinned image is used later for feature (minutiae) extraction. An example of the thinned image is presented in Figure 2.23.

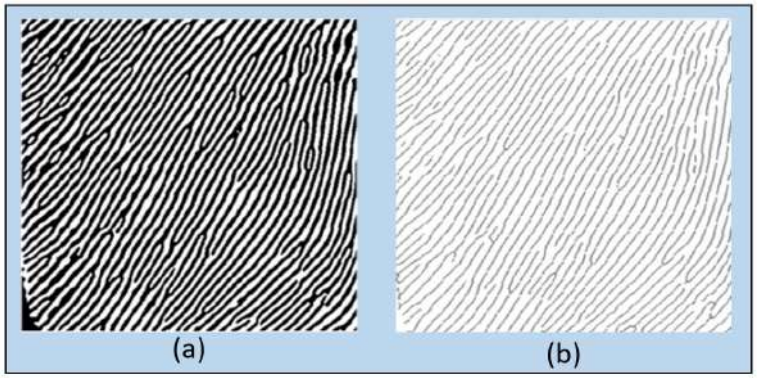

FIGURE 2.23: (a) Binary palm patch (b) Thinned palm patch

2.2.7 Feature Extraction

As explained in Table 1.1 in Chapter 1, palmprints provide multiple levels of features at multiple image resolutions. Level 2 features "Minutiae" are obtained at high resolution and are considered the most reliable as palmprint templates based on minutiae stored in the database cannot be easily faked or spoofed. Furthermore, minutiae have forensic value as well [7]. Another benefit of using minutiae points for identification is that they provide sufficient identification accuracy and do not need to be fused with other features to improve matching performance.
Minutiae were first observed in fingerprints by Galton [1] as discontinuities in ridge lines. Each minutia could be represented by its position (x, y) and orientation with respect to the x-axis θ. Initially, Galton identified seven different types of minutiae in fingerprints. Among them, ridge endings and bifurcations were found to be the most stable. Other minutiae types could be represented as a combination of ridge endings and bifurcations. That's why most minutiae extraction algorithms extract ridge endings and bifurcations only. Various types of minutiae identified by Galton are presented in Figure 2.24. Minutiae are generally extracted from thinned images [62]. Each ridge line is followed pixel by pixel and at each pixel,

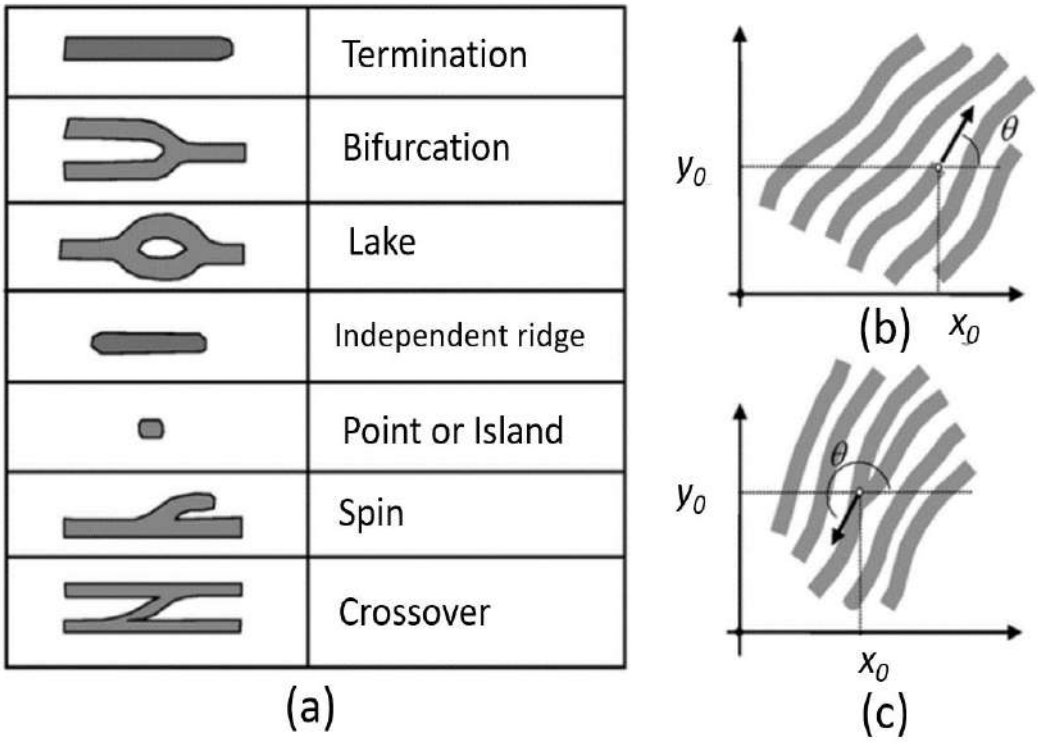

FIGURE 2.24: Minutiae types (a) table showing types of minutiae, (b) ridge ending or termination, (c) ridge bifurcation [1]

a local neighborhood of 3 is inspected. All 8 neighboring pixels are inspected in a clockwise direction and the number of transitions between black and white pixels is counted. If the number of transitions is 1 then it is a ridge ending. If the number of transitions is 3, then it is ridge bifurcation. This is illustrated in Figure 2.25.

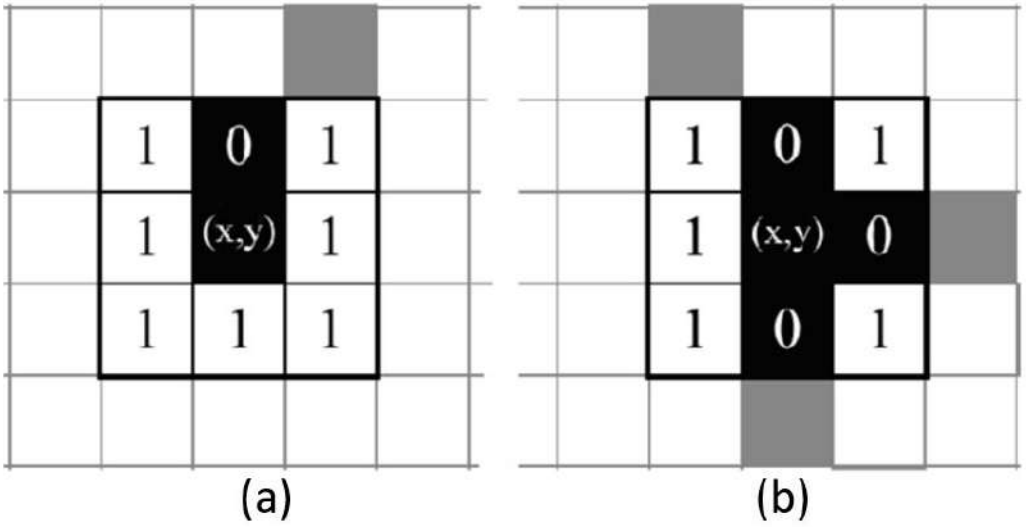

FIGURE 2.25: (a) Ridge ending: 1 transition from black to white (b) ridge bifurcation: 3 transitions

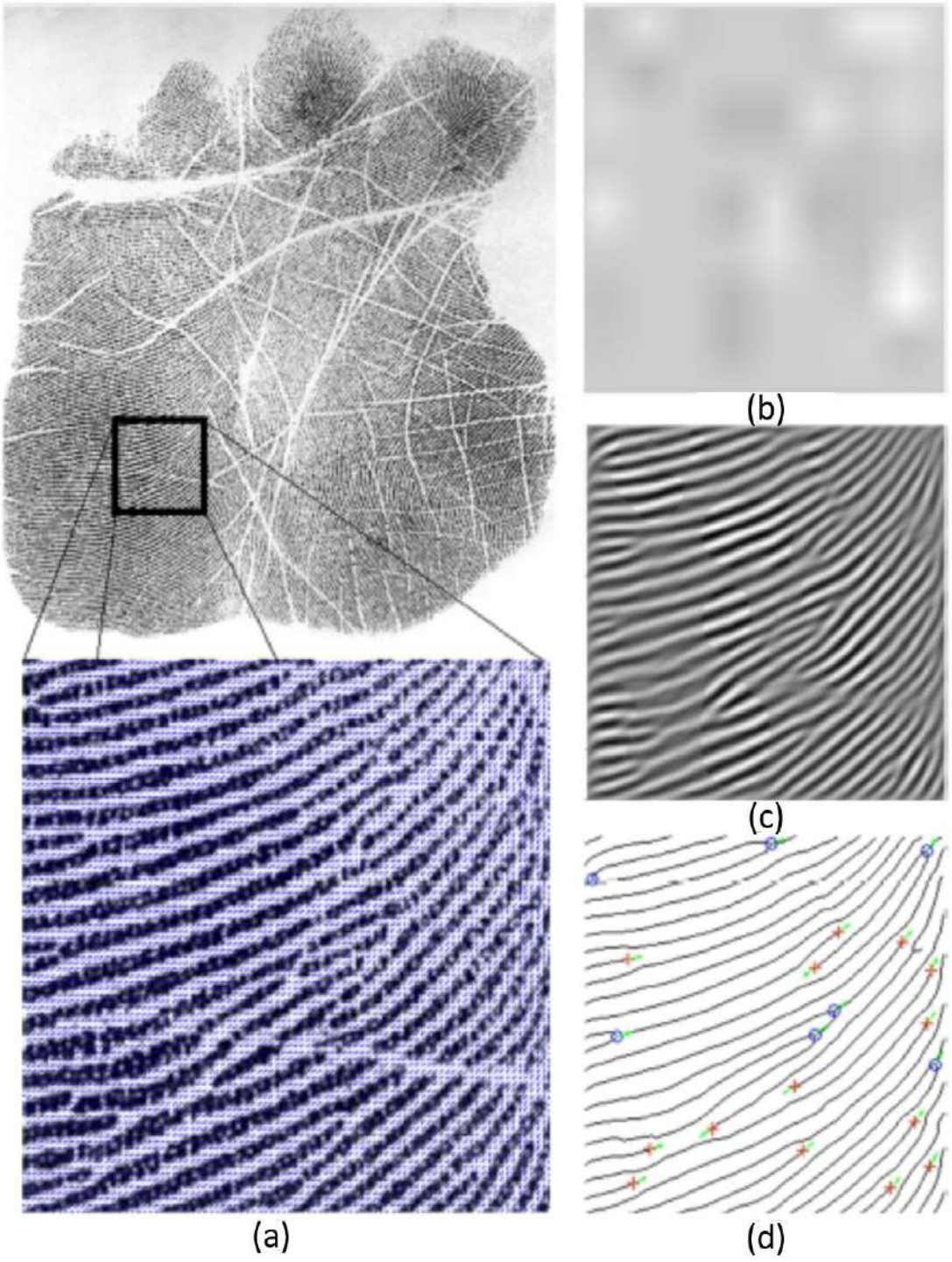

FIGURE 2.26: Complete palmprint processing steps: (a) Local orientation estimation in a patch depicted by small blue lines running along ridge lines, (b) ridge frequency map, (c) contextual filtering, (d) thinned patch with minutiae

The overall process described in section 2.2.1 through section 2.2.7 including estimation of ridge orientation, ridge frequency, contextual filtering, binarization and thinning, and minutiae extraction is illustrated with a practical example in Figure 2.26.

2.2.8 Dealing with False Features

Due to a large number of creases in the palm, most of the minutiae extracted after thinning are false. These minutiae arise due to discontinuities in the ridges resulting from scars, creases, or any other possible occlusions (Figure 2.27. Ridge endings at the boundary of a palmprint are also wrongly classified as minutiae by feature extraction algorithms. These false minutiae not only increase computational overhead in subsequent encoding and matching but also deteriorate identification accuracy by increasing the false acceptance rate (FAR).

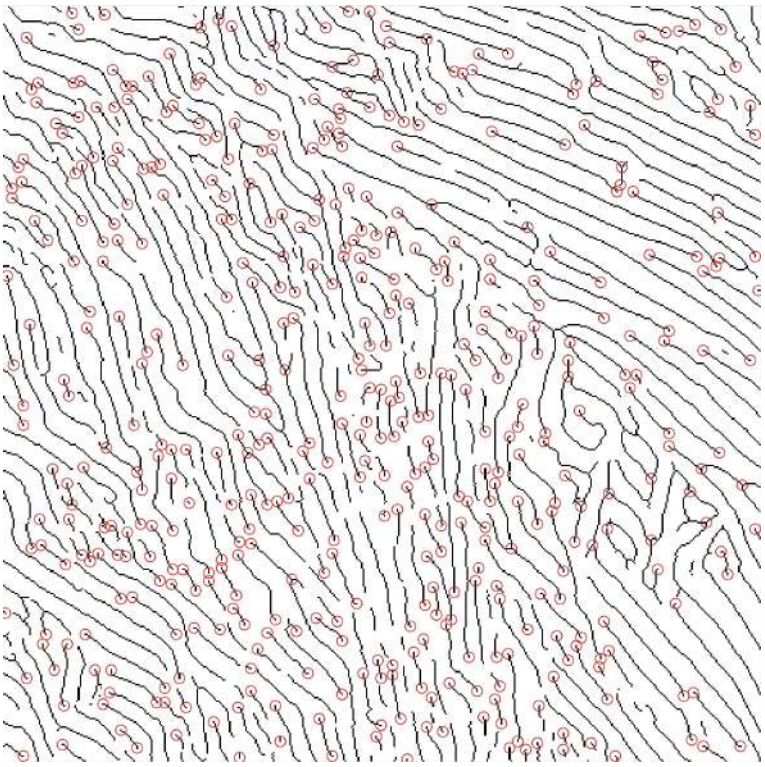

FIGURE 2.27: An example of palm patch with large number of false minutiae

A rudimentary way of eliminating these false minutiae is by rejecting minutiae from regions of poor ridge quality, rejecting minutiae that are too close to each other but have different orientations, or rejecting minutiae from regions with an extremely high density of minutiae because these minutiae depict broken ridges. But researchers [7, 48, 63–65] have tried to remove these minutiae using different statistical and structural methods. Chapter 6 delves deep into the issue of false minutiae and proposes a minutiae selection approach that selects reliable candidate minutiae for the matching stage. The proposed method is able to reduce

computations greatly by reducing the number of minutiae that enter the matching stage. An example of false minutiae is presented in Figure 2.27. Palmprint is of poor quality and is taken from THUPALMLAB dataset [44].

2.2.9 Feature Encoding

After the extraction of features, next step is to find a suitable representation for the features called encoding (Figure 2.8). Minutiae are represented by their location and orientation with respect to x-axis in the form (x, y, θ). This representation helps in uniquely identifying a minutia point. If palmprints are acquired in a controlled environment, they are aligned and (x, y, θ) representation is enough to match the minutiae points between a query image and stored templates. But in real scenarios, palmprints are not aligned and sometimes they are partial. Hence, using merely (x, y, θ) is not robust to translation and rotation.

Matching algorithms in such scenarios need special minutiae encoding mechanisms that are resilient to palmprint translation and rotation. In order to do that, encoding methods associate additional information to each minutia. Like other methods, minutiae encoding for palmprints has also been borrowed from fingerprints. In fingerprints, four types of information are attached to each minutia. These include: image intensity[66], texture[67], ridge characteristics [68] and neighboring minutiae[69, 70].

Encoding each minutia with respect to its nearest neighboring minutiae is the most popular method found in the literature. Neighbor-based minutiae schemes can be classified as fixed radius and k-nearest neighbors methods. In [7], Jain et al. proposed a fixed-radius method and divided a neighborhood around each minutia into 32 sectors and classified all neighboring minutiae based on their reliability and similarity of orientation with center minutiae. They called it *MinutiaCode*. They categorized types of neighboring minutiae as below:

- RS: Reliable and same orientation as the center minutia
- US: Unreliable and same orientation as the center minutia

- RO: Reliable and opposite orientation as the center minutia
- UO: Unreliable and opposite orientation as the center minutia

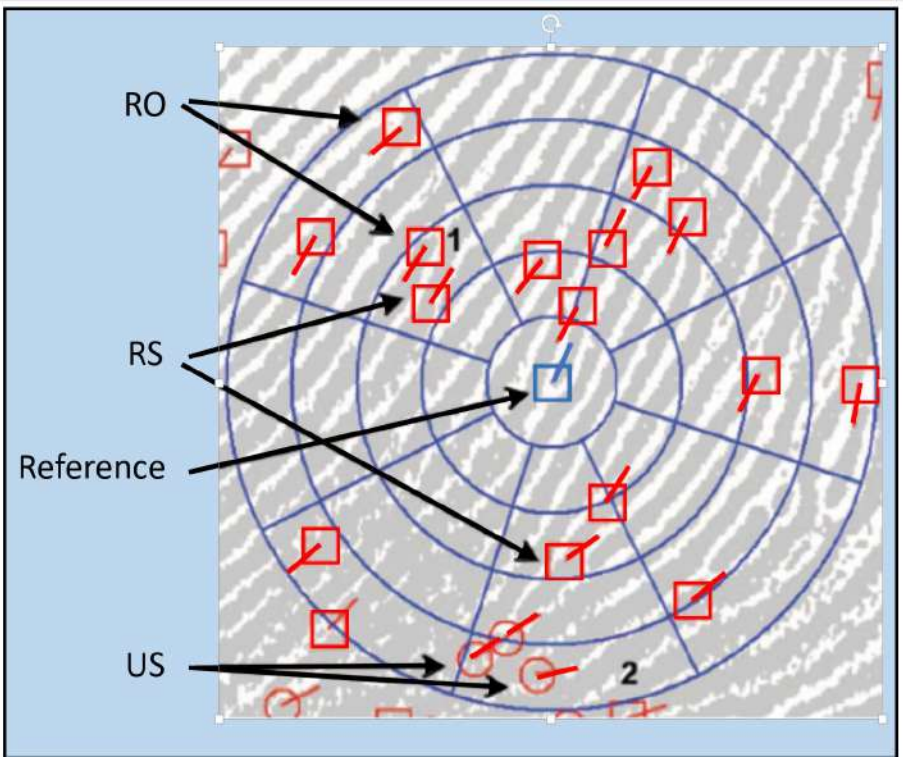

FIGURE 2.28: The configuration of a *MinutiaCode*. The numbers of four types of neighboring minutiae, RS, US, RO, and UO, in sectors 1 and 2 are [1 0 1 0] and [0 2 0 0], respectively. Square indicates reliable minutiae and circle indicates unreliable minutiae.

The similarity between two minutiae (query and template) is calculated as a weighted average of all valid sectors. If the score is more than 16, minutiae are deemed similar, otherwise not. This is illustrated in Figure 2.28. Same technique is improved by Dai et al [56] with the fusion of minutiae with other features. Another popular encoding scheme for minutiae encoding is Minutia Cylinder Code (MCC) [54]. MCC is also a fixed-radius local descriptor that is invariant to translation and rotation. It encodes each minutia in the form of a finite-sized binary vector. The fact that it is in the form of a bit vector makes it very efficient for the matching process. MCC is used in some recent works on high resolution palmprints [57, 58].

Another popular method of encoding minutiae is by taking the nearest k number of minutiae and estimating the difference of x-coordinate, y-coordinate and θ

from the reference minutia. These differences are calculated for all minutiae in a palm and saved as a template in the database. Suppose a palmprint consists of M number of minutiae, where each minutia is represented as (x, y, θ). Then for each minutiae, there exists a set P which contains differences of (x, y) and θ given by $P = \{(\Delta x_1, \Delta y_1, \Delta\theta_1), ...(\Delta x_k, \Delta y_k, \Delta\theta_k)\}$, where k is the total number of neighbours used for encoding. A similar set P exists for all M minutiae in a palmprint.

Jiang and Yau [71] proposed an encoding scheme based on neighboring minutiae using a polar vector. Polar vector-based encoding has been used in recent works [47, 48, 51]. Figure 2.29 illustrates a reference minutia $m_i(x_i, y_i, \theta_i)$ along with its two neighboring minutiae $n_0(x_0, y_0, \theta_0)$ and $n_1(x_1, y_1, \theta_1)$.

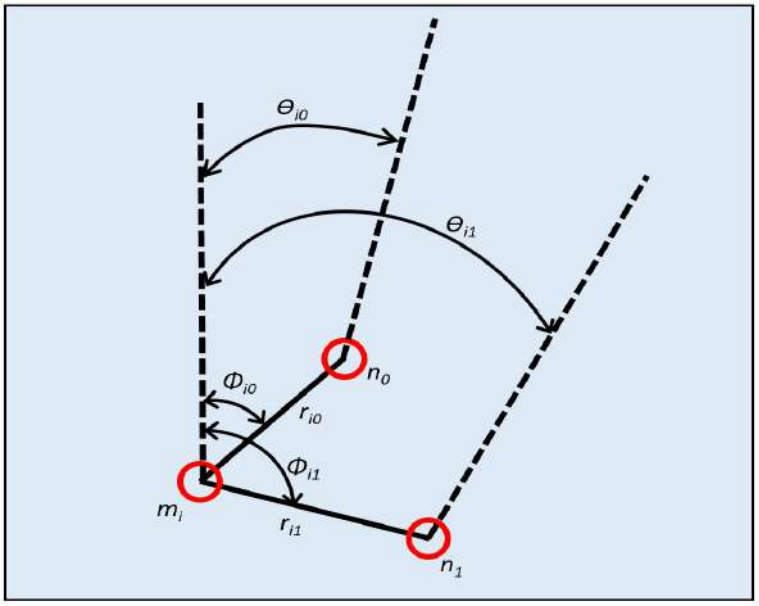

FIGURE 2.29: Minutiae encoding scheme based on triplet structure [71]

Polar vector for reference minutia m_i is described as

$$F_i = (r_{i0}, r_{i,1}, \theta_{i0}, \theta_{i1}, \phi_{i0}, \phi_{i1}, n_{i0}, n_{i1}, t_i, t_0, t_1), \tag{2.14}$$

where r_{i0} and r_{i1} are the distances of neighboring minutiae points n_0 and n_1 in polar coordinates from reference minutia m_i.

θ_{i0} and θ_{i1} are the differences in angles of neighboring minutia n_0 and n_1 and reference minutia m_i.

ϕ_{i0} and ϕ_{i1} are the difference between orientation of line sections joining m_i with

n_0 and n_1 and θ_i.

n_{i0} and n_{i1} are the number of ridges between m_i and neighboring minutiae n_0 and n_1, and

t_i, t_0, t_1 represent types of minutiae, i.e., ridge ending or bifurcation

2.2.10 Matching

Regardless of encoding techniques, minutiae-matching methods include an exhaustive search algorithm that compares each minutia from the query palmprint to each minutia in the template palmprint. Let there be M minutiae in query and N in template palmprint. Matching of a reference minutia m_i (where $i = 1, 2, ...M$) in query palmprint with every minutiae of template palmprint requires matching of each component of encoded set $P_i = \{(\Delta x_1, \Delta y_1, \Delta\theta_1), ...(\Delta x_n, \Delta y_n, \Delta\theta_n)\}$ for reference minutiae with all sets $R = \{R_1, R_2, ...R_N\}$ of template palmprint, where n is the total numbers of neighbours used to encode a single minutia. Process is illustrated in Figure 2.30. In order to match one query minutia with one

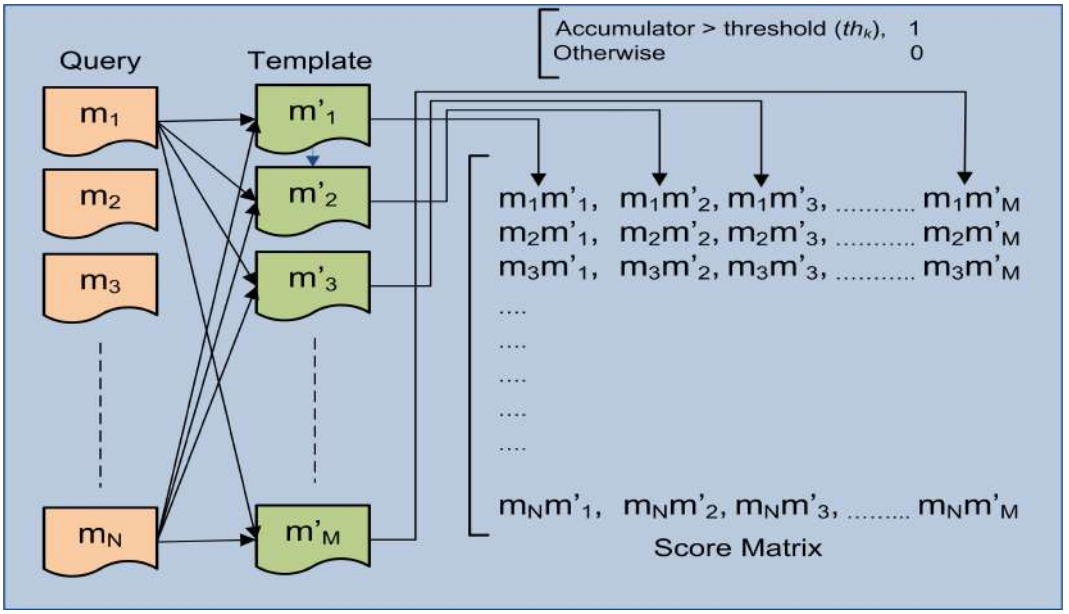

FIGURE 2.30: Process of matching minutiae of two palmprints [51]

template minutia, a total of n^2 matches are performed. This means total matches for two complete palmprints are $M \times N \times n^2$. Generally, the accuracy of matching increases with an increase in the number n but this comes at a great computational

cost given the large number of minutiae extracted from a full-size palmprint. As can be seen in Figure 2.30, based on matches a *score matrix* is calculated. For a minutia pair $(m_i, m_i^{'})$, if out of n neighbouring minutiae more than th_n number of minutiae match then query minutia m_i is said to have matched with template minutia $m_i^{'}$. th_n is the threshold of neighboring minutiae matches for a minutiae pair to match.

Although the generic method of minutiae matching remains the same, various improvements have been suggested by recent works. For example, in [48] Ghafoor et al. proposed a two-stage matching algorithm: Local and Global matching. Local matching is carried out similarly to the method described in the preceding paragraph. Later, the top 20 matched minutiae are selected based on the matching score calculated at the local level. These 20 minutiae are encoded again and matched in a similar fashion to achieve superior functionality.

Recent works also include the fusion of multiple biometric modalities for better performance. In [56], Dai et al. try different feature combinations of minutiae with ridge orientation map, ridge density map, and principal creases and show that fusion shows better accuracy at the cost of greater computations. In [58], Hussain et al., use an orientation descriptor around each minutia $m(x, y, \theta)$ encoded using Minutia Cylinder Code (MCC). Before MCC matching between two minutiae points, orientation descriptors are matched. If they are similar enough, then the MCC matcher is used otherwise the match is discarded. This saved computation during matching.

Convolutional neural networks (CNNs) have also seen remarkable success in the image processing field in the last decade. Due to that success, there is a rise in the use of CNNs in palmprint identification as well. However, the use of CNNs in palmprints is more frequently seen in low resolution palmprint identification [72–78]. This is because high resolution palmprints have a large region of interest that is hard to be processed by a CNN in a single attempt and region-wise processing of palmprints through CNNs has its associated computational cost. Additionally, the large size complicates network training for the extraction of level 2 features. Due to these reasons, limited application of CNNs for high resolution palmprints is found in the literature. Ahmadi and Soleimani [79] applied CNNs on high resolution

palmprints to roughly predict palmprint rotation around *z-axis* and subsequently used Generalized Hough Transform (GHT) to find exact rotation and translation difference between query and template palmprint for image registration. Image registration brings both palmprints to the same coordinate systems which helps in speeding up the matching process. Fanchang, Hao, et al. [15] used CNNs to classify multiscale high resolution palmprint patches according to the quality of ridge structure contained in them. A more recent work employing CNNs for high resolution palmprints is presented by Bing and Feng [13] who employed Generative Adversarial Networks (GANs) to estimate ridge orientation only. Limited application of CNNs on high resolution palmprints shows that CNNs are more suitable for palmprint identification based on low level features such as principal lines.

2.3 Research Gap Analysis

The topic of palmprint identification comprises an elaborate amount of literature. Research spans multiple facets of palmprint systems like enhancement, finding reliable features, feature encoding, and matching. Most of these methods have been borrowed from fingerprints and customized for palmprints. Recent works on palmprint systems have focused more on improving the efficiency of matching algorithms in terms of accuracy and computational cost. It is highlighted that the accuracy achieved in matching is contingent upon the success of ROI segmentation, enhancement, and selection of good minutiae candidates.

Research Gap-1: Due to the large size of the palmprint, a robust ROI segmentation is required that does not confuse background textures with palmar regions of the image. variance-based ROI segmentation methods used in fingerprints previously are not able to work on palmprints efficiently. This is due to the fact that palmprints acquired from real-world scenarios contain complex background textures which can have the same textural properties (e.g. variance) as the palmar regions in the image. Due to this results of ROI segmentation are not efficient resulting in increased computational requirements and extraction of false features from the background pixels.

Research Gap-2: It can be seen in Table 2.1, that limited novelty has been introduced in the palmprint enhancement. In order to recover palm ridge structure, most enhancement methods employ contextual filtering either in the spatial domain or frequency domain. In palmprints, contextual filtering gives unreliable results in high crease areas. This is because:

- Gradient-based spatial methods mistake palm creases as ridge lines as maximum intensity change in pixel values is provided by creases rather than ridge lines in a high-crease region of the palmprint.
- All frequency domain methods assume the underlying ridge patterns to be stationary or uniform in a local patch. This is a risky assumption in the case of palmprint due to abruptly changing ridge patterns and a high number of creases. For the same reason, the size of the patch has to remain small to provide a more accurate estimate of local ridge orientation and frequency.

Research Gap-3: Limited effort has been rendered in eliminating false minutiae from the palmprint after the feature extraction phase. This results in an unnecessarily high number of minutiae being forwarded to the matching stage. A large number of false minutiae in the matching stage results in poor matching accuracy as well as unnecessarily increased computational overhead.

2.4 Problem Statement

Since palmprints acquired from real-world scenarios contain complex background structures, there is a need to develop an ROI segmentation method that isolates only the pixels containing ridge lines in the image regardless of the variance or other statistical properties.

In order to overcome the problem of a high number of creases present in the palmprint, there is a need to find an enhancement method that is able to recover the underlying ridge structure with adequate robustness even in high crease areas. This requires the enhancement method to be adaptable to abruptly changing

ridge orientations, discontinuities introduced due to creases, the flexibility of palm skin, poor ridge/valley contrast, and background noise. The enhancement method should also be able to work on relatively larger patches of palmprint without assuming underlying ridge patterns to be stationary or uniform.
To improve matching efficiency in terms of accuracy and computational cost, there is a need to devise a minutiae selection algorithm that eliminates false minutiae after feature extraction and shortlists good candidates for subsequent feature encoding and matching. Algorithms should be able to improve accuracy as well as computational efficiency.

2.5 Objectives of the Thesis

To summarize, high resolution palmprints provide the following challenges:

- Poor contrast between foreground pixels (palmar area) and background pixels (non-palmar area) of the image
- Large amount of minor creases resulting in abrupt changes in ridge orientation
- flexibility of skin in the palm resulting in the inconsistent inter-ridge distance.
- Large size of palmprint resulting in computational overhead during enhancement and matching phases.

The objective of this thesis is to develop an ROI segmentation method that isolates pixels containing palmar ridge lines only and a palmprint enhancement method that is able to adapt to a multitude of challenges offered by palmprints acquired either in a controlled environment or from real-world scenarios like crime scenes. The goal of the enhancement method is to recover the ridge structure of the palm despite frequently appearing creases, the flexibility of the skin, and poor contrast of foreground and background pixels.
Secondly, this study aims to develop a minutiae filtering method for speeding up

the matching or identification process. Since palmprints are large and provide 8 times more minutiae than fingerprints, it is essential to limit matching to only good quality features which not only speeds up the matching process but also improves accuracy.

2.6 Summary

This chapter presents a comprehensive literature survey on palmprint identification systems ranging from low to high resolution palmprints. It lists the features used in both low and high resolution palmprints and possible applications of both. Chapter dives deep into high resolution palmprints and provides details about the challenges involved in designing a high resolution palmprint system and the importance of the efficacy of the palmprint enhancement algorithm. Processes reviewed include ROI segmentation, ridge orientation, and frequency estimation, contextual filtering, feature extraction, encoding, and finally matching.

Palmprint enhancement processes are more elaborately reviewed as they are the prime focus of this thesis. It is reiterated that identification accuracy is contingent upon feature enhancement and extraction. Removal of background textures from the palmprint and extraction of reliable ridge patterns is essential for post-processing to succeed. In the next chapter, a novel frequency-based ROI segmentation method is proposed as a precursor to subsequent palmprint enhancement. Instead of finding textual differences, it directly targets ridge patterns in the image to extract only the foreground pixels. Later a palmprint enhancement method is proposed for those by-passes traditional methods and recovers ridge structure using deep learning techniques.

Chapter 3

Proposed Region of Interest (ROI) Segmentation Method

3.1 Background

Fourier Transform is a popular tool for breaking down a signal and expressing it as a sum of sinusoids, where each sinusoid represents a particular frequency in the signal. Images can be thought of as two-dimensional (2D) signals with numerous frequency components. So by extension of the same concept, a 2D Fourier transform gives information about the frequencies contained in an image. In most applications, the 2D Fourier transform is re-arranged to shift zero frequency $F(0,0)$ or the DC offset to the center of the transform. In the frequency domain, different frequency components contained in an image are represented as peaks at varying distances from the center point, i.e. f_0. The higher the frequency, the farther it is from the center point. Each frequency is represented by two peaks on opposite sides of f_0 because the Fourier transform is symmetric around the center. Furthermore, a spatial frequency component's corresponding peak in the frequency domain contains associated orientation information also.

Figure 3.1 presents a generic example of how spatial frequencies are represented in the frequency domain. It shows two 2D sinusoidal waves: (A) is a low frequency sinusoidal wave having zero angle with respect to the x-axis, while (B) is a

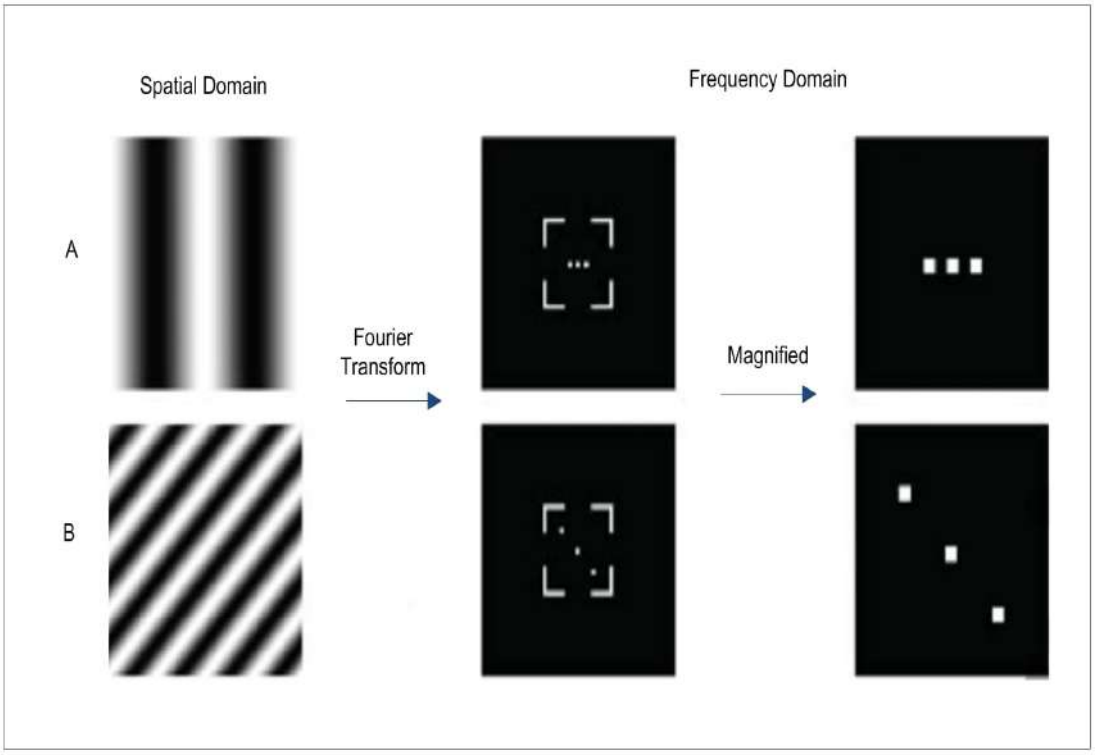

FIGURE 3.1: 2D Fourier Transform example: (A) sinusoidal wave with low frequency at zero angle with respect to the x-axis, (B) sinusoidal wave with higher frequency at an angle with respect to the x-axis. The left column shows the spatial domain, the middle column shows the frequency domain and the right column is a magnified version of the frequency domain

high frequency sinusoidal wave at a certain angle with respect to the x-axis. Low frequency wave (A) is represented by two peaks in the frequency domain which are closer to the center of the spectrum. Whereas, higher frequency wave (B) is represented by two peaks that are farther from the origin. Secondly, peaks are oriented according to the orientation of spatial frequency. The amplitude of both sinusoidal waves is represented by the brightness of peaks in the frequency domain. The alternating ridge/ valley pattern in the palmprints can be thought of as a 2D sinusoidal wave. The orientation and frequency of this sinusoidal are different in different parts of the palm but the generic pattern remains constant. Figure 3.2 shows how the local palm ridge structure can be modeled as a suitable 2D sine wave. The Fourier transform of palmprints that are extracted from real scenarios contains a massive amount of frequency components corresponding to objects other than the ridge pattern. It contains low frequency components corresponding to background textures, higher frequency components corresponding to noise, and frequency components corresponding to the ridge pattern.

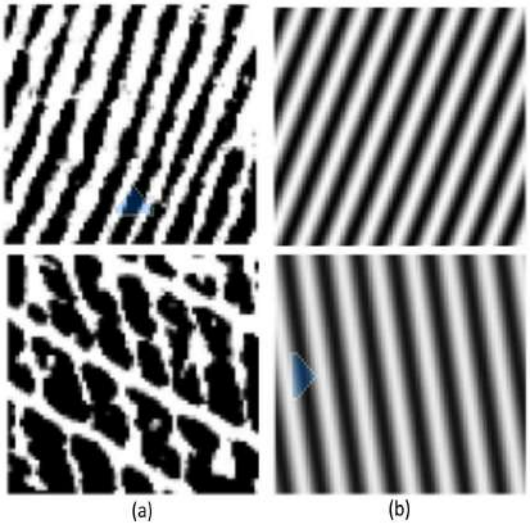

FIGURE 3.2: Ridge structure modeled as 2D sine waves: (a) palm patch containing only ridges, (b) palm patch containing ridges and creases.

3.2 Proposed ROI Segmentation

The whole purpose of ROI segmentation is the extraction of the region of interest in the image which in our case is the area containing the palm ridge structure. Due to a constant sinusoidal pattern of ridge structure, it is possible to isolate ridge frequency in the frequency spectrum and discard all others. Hence, the proposed ROI segmentation is carried out in the frequency domain. The overall process of proposed segmentation is explained in Figure 3.3. Palmprint is converted into its Discrete Fourier Transform (DFT) equivalent and multiplied by a bandpass filter that is configured to only allow frequencies corresponding to the ridge pattern. After multiplication, the palmprint is brought back into the spatial domain. Heavy blurring is used to smooth the image. subsequently, the image is binarized using a threshold to yield a binary mask that is multiplied by the original image to output only the ridge-containing regions of the palm.

The bandpass filter is a combination of low and high pass filters. It attenuates all frequencies below and higher than an allowed band of frequencies. The bandpass filter is particularly useful in enhancing edges while suppressing background (low frequency components) and noise (high frequency components). Since ridge lines are quite like edges, a bandpass filter suits our application. Ridge lines show different orientations across the palmprint. Secondly, inter-ridge distance or ridge

frequency keeps changing on account of the flexibility of the skin.

Ridge frequency changes between 9 to 11 pixels in a 500 ppi image [7](which is the most common resolution of high resolution palmprints). Ridge frequency f_{ridge} can be easily isolated in the DFT of the image. Maximum frequency f_{max} in an image is the one which has a wavelength of just 2 pixels, i.e., it shifts from high to low and back to high intensity in just 2 pixels. DFT of an image depicts frequency components in such a way that the higher the spatial frequency, the farther it is from the center (f_0). This implies that a spatial frequency with a minimum wavelength of 2 pixels (λ_{min}) along the x-axis will be represented by a peak (f_{max}) on the far right of the x-axis in DFT.

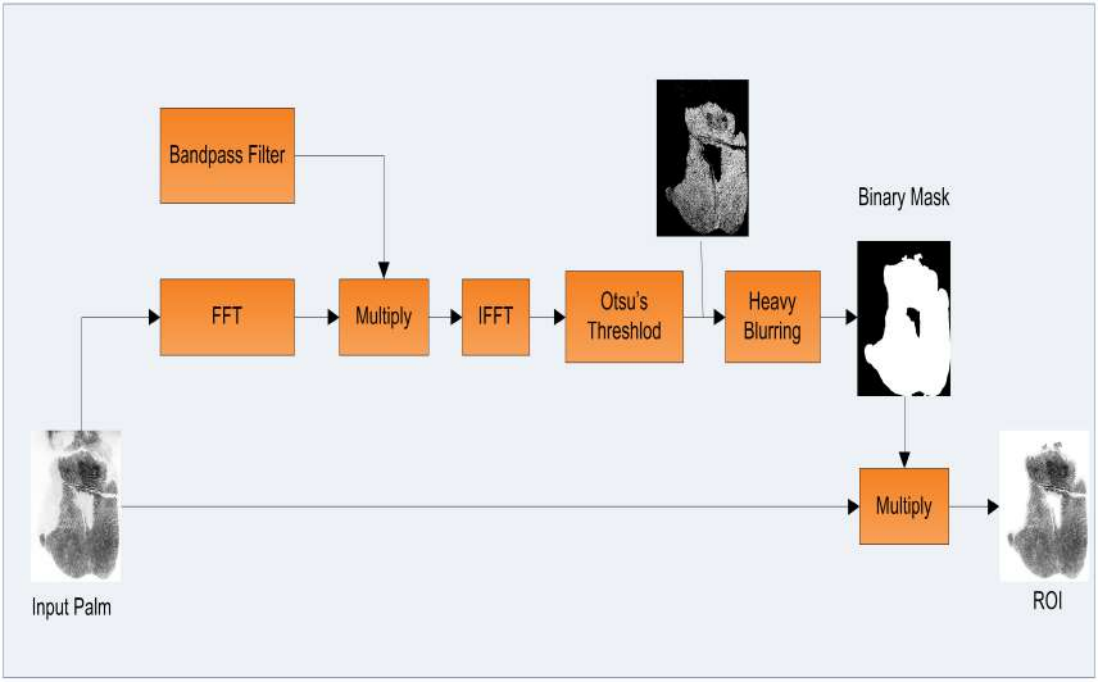

FIGURE 3.3: Proposed ROI segmentation: Palmprint is multiplied by a band-pass filter in the frequency domain to remove noise. In the spatial domain, Otsu's thresholding is used to binarize the image and heavy blurring is used subsequently to yield a binary mask which is multiplied with the original palm-print to yield only the foreground pixels.

Since the image and its DFT have the same dimensions, the highest frequency f_{max} along the x-axis in an $N \times N$ image will be at a distance (d_{max}) from the center (f_0). Based on this conclusion we can find the approximate distance of ridge frequency, f_{ridge} ($1/\lambda_{ridge}$, where $\lambda_{ridge} = 10$ pixels [7] from the center (f_0)

in a 500 ppi image Distance of f_{max} from the origin is given by

$$d_{max} = \frac{N}{2}. \tag{3.1}$$

Now since f_{max} is at a distance of d_{max} $(N/2)$ from the centre, d_{ridge} can be calculated as:

$$d_{ridge} = \frac{N}{2} \times \frac{2}{10}, \tag{3.2}$$

$$d_{ridge} = \frac{N}{10}. \tag{3.3}$$

In the proposed implementation, a Gaussian bandpass filter has been used to extract ridge frequency which is defined as:

$$H(u,v) = exp\left[-1/2\left[\frac{D^2(u,v) - D_o^2}{D(u,v)W}\right]^2\right], \tag{3.4}$$

where $D(u,v)$ is the distance of each frequency component from the center and D_o is the radial distance of the band center from the DFT image center. W is the width of the filter and defines the Gaussian spread in both directions. While dealing with 2040×2040 images, bandpass filter with a D_o at 180 pixels with bandwidth W of 50 was found to be effective in catering for variations in ridge frequencies and ridge orientations. DFT analysis of palmprint is presented in Figure. 3.4.

It can be seen in Figure 3.4, that the DFT of palmprints shows a distinct ring of frequencies that corresponds to ridge lines. In the middle of the DFT image, there are very luminous low-frequency components that correspond to the background in the image. While higher frequency components on the outside of the ring correspond to noise. Bandpass filter discards both low and high frequencies. Once filtered, the image is brought back into the spatial domain using Inverse DFT. At this stage, due to the removal of DC offset and low frequency components, the image is dark except where the ridges are. Ridge lines at this stage have varying intensity values. In order to make a distinct difference between ridges and the

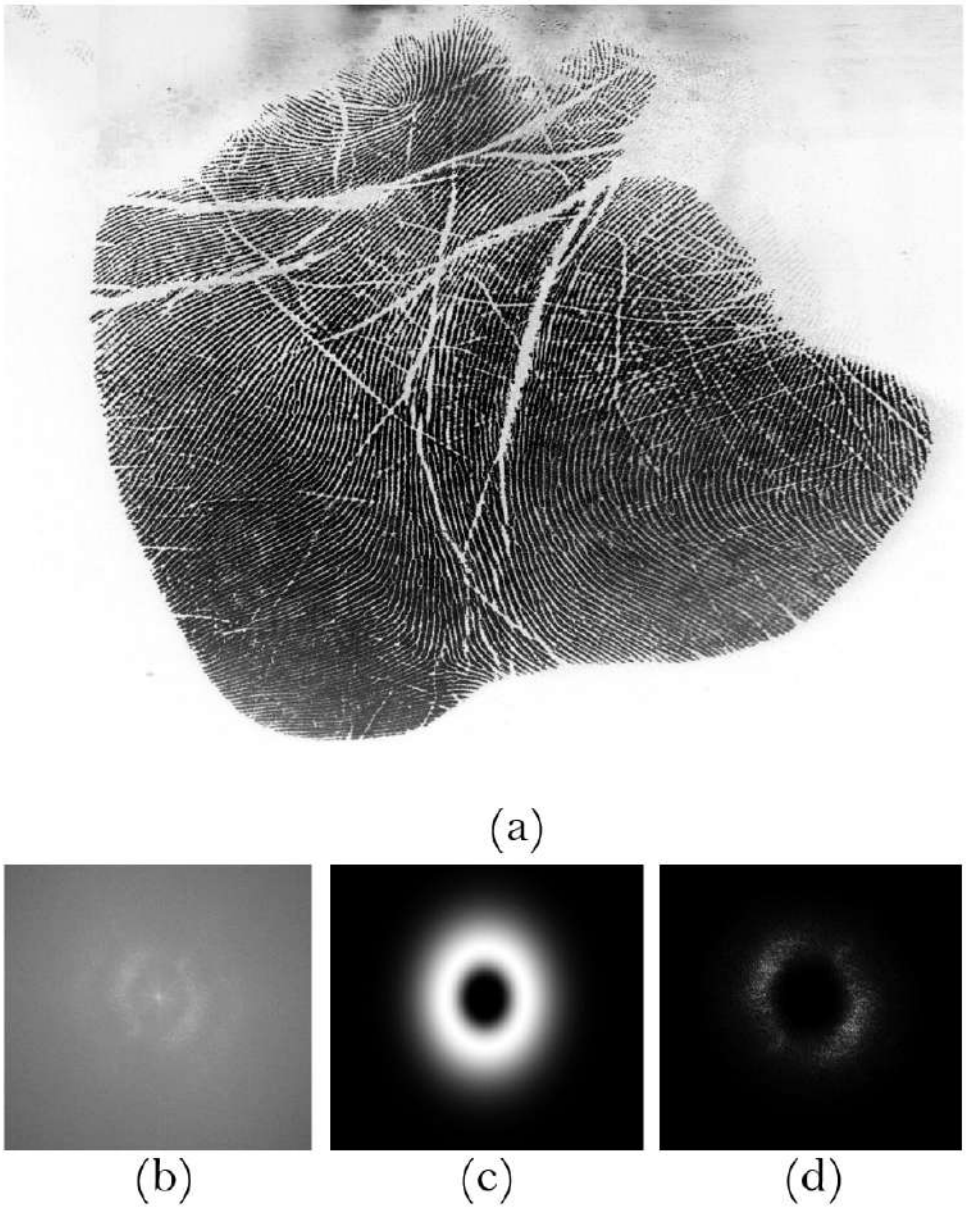

FIGURE 3.4: DFT Analysis of palmprint, (a) Input palmprint, (b) DFT of palmprint, (c) Bandpass filter, (d) Filtered palmprint in frequency domain

background, the image is binarized using a threshold obtained by Otsu's method. Otsu's method aims at finding the threshold that minimizes the inter-class variance between two classes of pixels that are separated by the threshold. The image obtained after applying Otsu's method is then smoothed through heavy blurring using a Gaussian filter with a standard deviation set at 20. This gives a binary mask which is subsequently multiplied by the input palmprint to reveal only the foreground pixels (ROI) of the palmprint. Results of multiplying binary mask with input palmprint are illustrated in Figure 3.5.

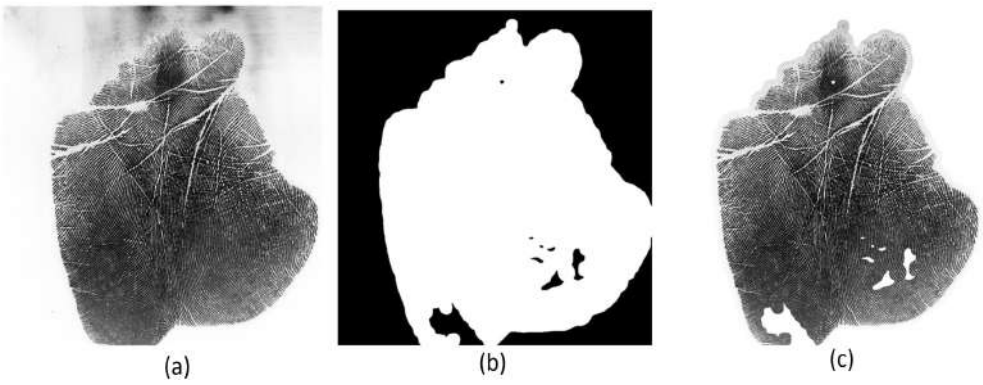

FIGURE 3.5: (a) Original palmprint, (b) Binary mask, (c) Segmented palmprint

3.3 Results

Some results of the proposed ROI segmentation method on a variety of image qualities are presented in Figure 3.6.

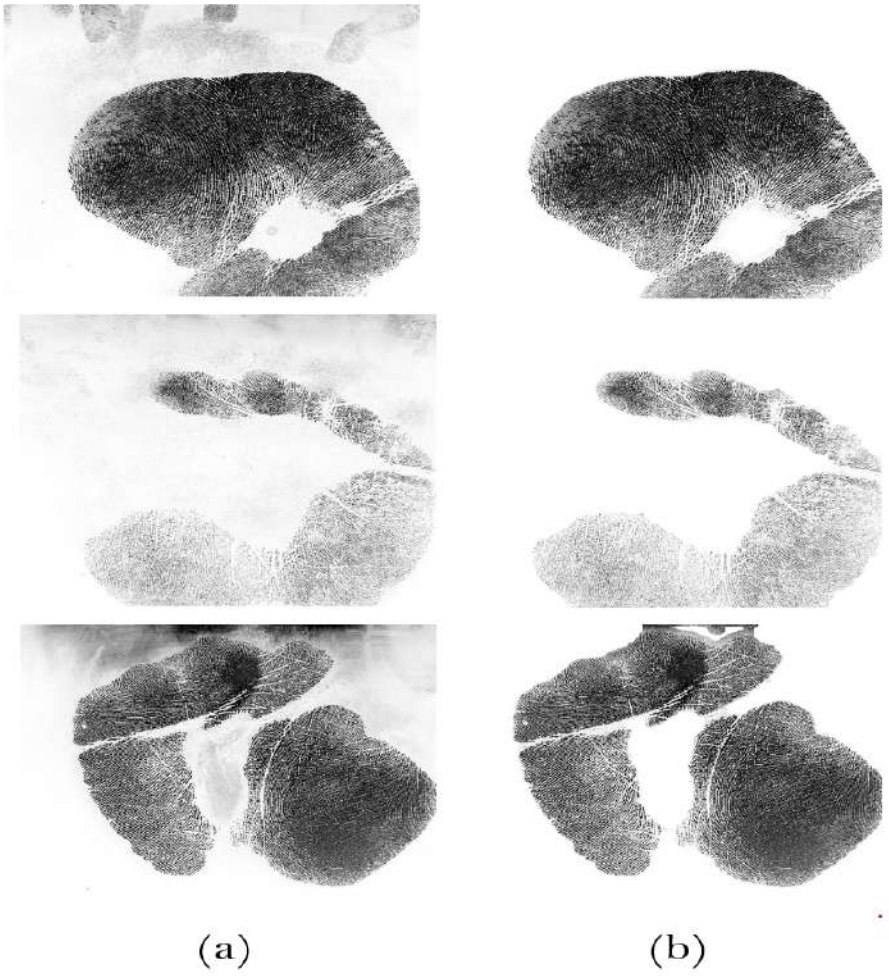

FIGURE 3.6: Proposed ROI segmentation results: (a) Original palmprints, (b) Segmented palmprints

Figure 3.6 depicts that the proposed ROI segmentation is able to extract the foreground by removing low frequency background and high frequency noise in the palmprint.

3.3.1 Comparison with Variance-based Segmentation Methods

Segmentation methods used in fingerprints [46, 62, 71] and palmprints [47, 48, 51] usually work by estimating local variances in the image. Area with ridges usually has a higher variance of gray scales due to alternating ridge/valley structure than background textures. A threshold on local gray scale variance is used to differentiate foreground from background pixels. The problem with this approach is that palmprints extracted from real-world scenarios such as crime scenes have very complex structures in the background whose variance can match the variance of regions containing ridge pattern. In such cases, segmentation results can include background pixels as well. This causes extra computation during all subsequent steps of palmprint enhancement and identification. It also can adversely affect identification accuracy as false features can be extracted from the background that might affect the false acceptance rate (FAR). A comparison of the proposed segmentation method with variance-based method is presented in Figure 3.7.
It can be seen the proposed segmentation method produces much sharper segmentation masks. Variance-based method misclassifies some background portions of the image as foreground due to the similarity in variance with foreground pixels. These portions are identified with red arrows in Figure 3.7. Whereas, the same portions are filtered out by the proposed segmentation. Another advantage of using the proposed method is that one filter can be used globally for the image. On the other hand, in variance-based methods, the variance threshold has to be adapted according to small neighborhoods, which is computationally exhaustive.

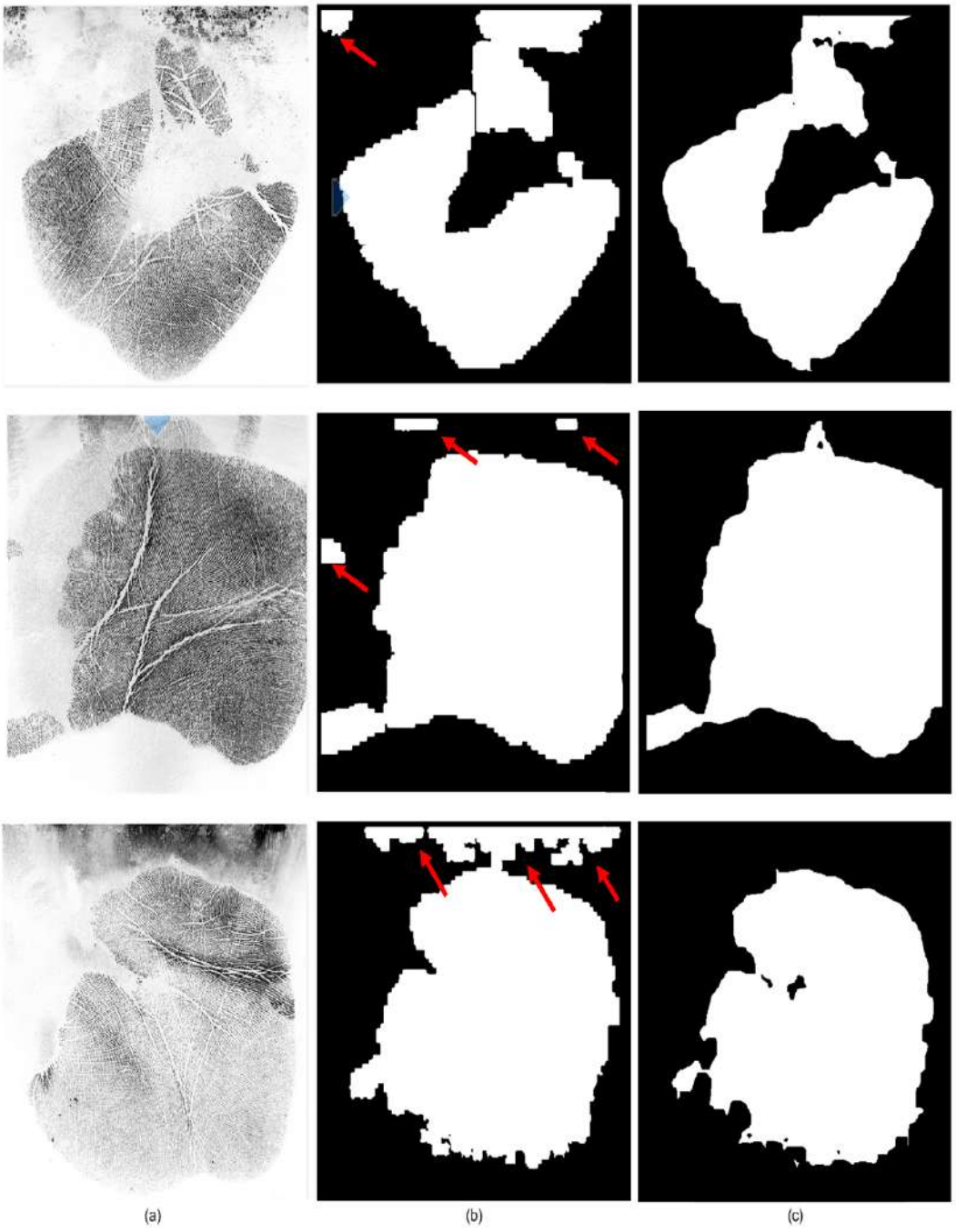

FIGURE 3.7: Comparison between proposed segmentation and variance-based segmentation: (a) Original palmprints, (b) mask created through the variance-based method, (c) mask created through proposed segmentation

3.4 Post processing

Sometimes palmprints acquired from real-world scenarios suffer from incompleteness. This is due to the non-uniform application of pressure on surfaces by a subject's palm. This can also happen in controlled environments where a subject

might not place his/ her palm properly on a scanner during the enrollment stage of verification. The flexibility of palm skin coupled with the complex physiology of the palm also contributes to this issue. In such cases, palmprints might contain some disconnected regions. It is found that features extracted from these disconnected regions are not reliable. Hence, disconnected regions that are small and do not provide any reliable features are removed as a post-processing step in the proposed segmentation method.

Similarly, if the palmprint images contain impressions of fingers as well, then fingerprints are not automatically segmented out by the proposed segmentation method. This is because fingerprints also have the same ridge structure as palmprints having the same frequency. In these cases, fingerprints have to be removed using post-processing methods because features extracted from the fingerprints cannot be matched with features stored in palmprint templates. An example of such a palmprint is provided in Figure 3.8

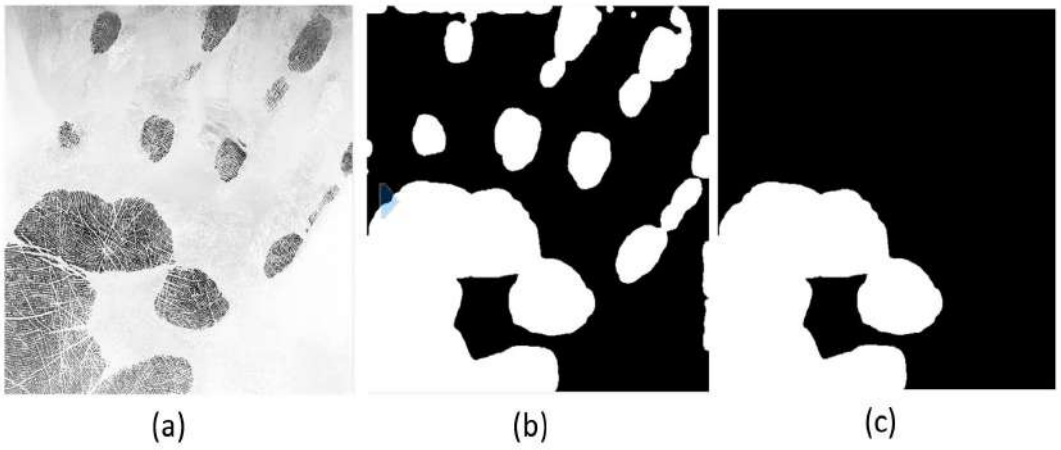

FIGURE 3.8: (a) Palmprint with associated fingerprints, (b) Segmented palmprint with fingerprints, (c) Segmented palmprint after post-processing

3.5 Summary

This chapter presents the proposed frequency domain ROI segmentation method which exploits the sinusoidal quality of ridge lines in the palmprint. Using a bandpass filter, the frequency spectrum of palmprint is filtered. Filtering removes all

low level frequency components pertaining to background and high level frequencies pertaining to noise. Palmprint is brought back into the spatial domain and binarized in which ridges are shown as white lines while the background is black. Heavy blurring is used on this palmprint to convert it into a binary mask. Mask is consequently multiplied with the original palmprint to yield only the foreground pixels. Subsequent enhancement, feature extraction, and matching are performed on the segmented image.

Chapter 4

Proposed Palmprint Enhancement Network (*PEN*)

4.1 Background

As stated in Chapter 2, palmprint identification systems consist of image enhancement, feature extraction, and matching stages. Palmprints cannot be used directly for identification and need to be enhanced to extract reliable and unique features. Performance achieved in enhancement dictates performance achieved during identification. High resolution palmprint enhancement techniques try to recover the palm ridge structure that may be degraded in various ways. This is a challenging task because most palmprints found in nature suffer from multiple degradations such as incompleteness, poor ridge/valley contrast, broken ridges, and the addition of external noise such as stains or background texture. Hence, it is imperative for enhancement methods to be adaptive to variable image qualities. Enhancement generally consists of the following processes:

- Estimation of local ridge orientation resulting in an orientation map for complete palmprint
- Estimation of local ridge frequency resulting in a frequency map for complete palmprint

- Contextual filtering done in a local fashion guided by ridge orientation and frequency maps.

Due to the similarity in features, most palmprint enhancement methods are borrowed from fingerprint enhancement methods, but palmprints differ from fingerprints in a few aspects, namely:

- Large number of creases resulting in abrupt changes in ridge orientation
- Flexibility of skin resulting in abrupt changes in inter-ridge distance or ridge frequency
- Palmprints are much larger and contain much more information which increases computation complexity

Due to the above-mentioned differences, applying fingerprint enhancement methods directly, or use of commercial software development kits (SDKs) designed for fingerprints does not give satisfactory results for palmprint enhancement. A major reason for this is the presence of a large number of major and minor creases in the palmprint. Due to these creases, an error-prone orientation and frequency map is calculated which subsequently results in the wrong configuration of contextual filters. As a result, the exact underlying ridge pattern cannot be recovered.
The most important step in palmprint enhancement is the estimation of local ridge orientation. Most popular high resolution palmprint enhancement methods found in the literature (Chapter 2) can be divided into two categories based on techniques used for the estimation of local ridge orientation. To summarize:

- **Gradient-Based Methods**: By far the most popular technique for enhancement. Local ridge orientation is estimated by calculating pixel-wise gradients, whereas ridge frequency is estimated using *x-signature* method described in section 2.2.4. Enhancement filters which are usually Gabor filters or other contextual filters, need to adapt their orientation and width according to local ridge orientation and frequency respectively. Once local ridge orientation is estimated for the whole image, filtering can be applied

in spatial domain [47, 48, 50, 51, 54, 55] or frequency domain [52, 53]. In the spatial domain, filtering is applied pixel-wise, where filter orientation and width are adapted to the ridge pattern in a small area centered at a pixel (x, y). In the frequency domain, filtering is applied patch-wise by performing a Fourier analysis of a patch of palmprint and multiplying with a suitable directional filter: Gabor or a raised cosine filter (Figure ??).

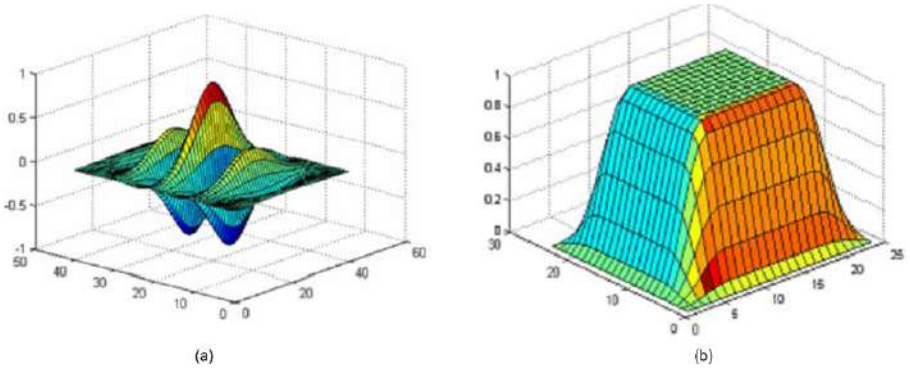

FIGURE 4.1: (a) Spatial domain Gabor Filter, (b) Frequency domain raised cosine filter [52]

These techniques work well in fingerprints but cannot handle abruptly changing ridge orientation and non-uniform ridge frequency resulting from flexibility of skin [53]. This results in an inexact configuration of contextual filters and error-prone enhancement.

- **Region-growing Methods**: Seeing limitations of the gradient-based estimation of local ridge orientation on palmprints, a frequency domain iterative region growing method was proposed by Funada et al. [80] that estimates local ridge orientation and frequency simultaneously. Assuming the Local ridge pattern to be constant in small non-overlapping patches (usually 8×8), they are modeled as $2D$ sine waves. In the image formed by the strongest sine wave in each patch, continuous blocks are joined together to form regions. Depending on various properties, regions are classified as *seed* or *crease* regions. A region-growing algorithm is used to grow seed regions until a complete orientation map of palmprint is produced. Subsequently, contextual filters are applied to the image in a local fashion whose orientation and width are guided by the orientation map. This technique was used

by Jain et al. in [7] and improvised by Dai et al. in [56] to speed up the process. Some recent works have also used this method for orientation and frequency estimation [57, 58, 61].

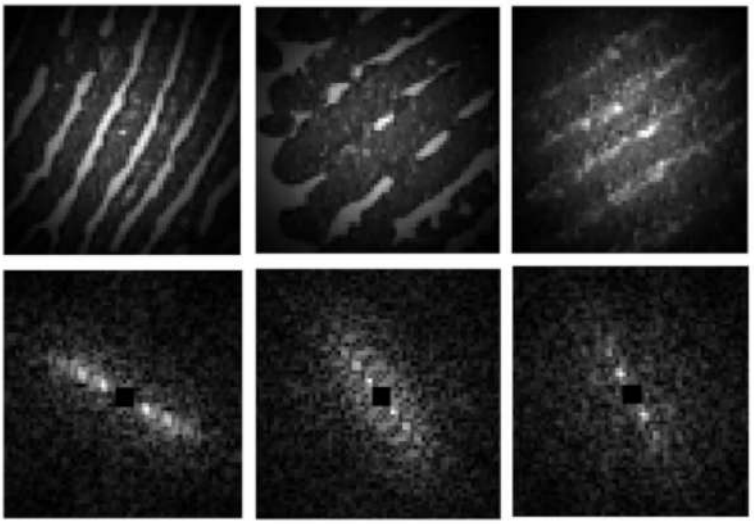

FIGURE 4.2: First row shows the original palmprint blocks and the second row shows their DFT amplitude showing multiple peaks [57]

Contrary to Figure 4.2, in poor-quality regions containing a lot of creases or background textures, the peak corresponding to ridgelines is not very clear and multiple peaks are detected in DFT. In such regions, the strongest frequency component detected by Fourier transform corresponds to creases rather than ridge lines which introduces errors in orientation estimation.

Pixel-wise estimation is more robust to abrupt changes in ridge orientation and frequency but proves computationally costly in the case of palmprints. And patch wise frequency domain operations work on the assumption that underlying ridge orientation and ridge frequency in a local area are stationary or uniform. This assumption may be convenient in fingerprints but in palmprints can lead to misleading results. Regardless of the methods used for orientation and frequency estimation, final enhancement is done using contextual filters which are configured according to ridge orientation and frequency estimates. However, in local areas with high creases, conventional techniques pick up contextual information pertaining to creases and end up enhancing creases rather than ridges. There is a need to find an enhancement technique that is able to process sufficiently

large patches without making the assumption that underlying ridge orientation and ridge frequency are stationary or uniform.

4.2 Image-to-Image Regression Capabilities of Convolutional Neural Networks (CNNs)

The last two decades have seen a remarkable success of CNNs in classification problems. CNNs have outperformed classical classification models on most benchmarks. Contrary to image classification tasks which predict a single label for the whole image, image-to-image regression extends the function of CNNs to pixel-wise prediction. Semantic segmentation was the first application to use CNNs for pixel-wise prediction [81, 82]. These applications used VGG [83] or ResNet [84] as base architectures and customized them by introducing skip connections, deconvolutional layers, etc. to facilitate the reconstruction/ restoration of input/ output correspondences. Most of these architectures are task-dependent. This chapter proposes a novel palmprint enhancement method that is inspired by recent works in image restoration and segmentation employing pixel-to-pixel learning in an end-to-end fashion using Convolutional Neural Networks (CNNs) [85–89]. These models have achieved great success but have not been employed specifically for the problem of palmprint enhancement.

Palmprint Enhancement Network (*PEN*) is a two-step enhancement framework consisting of a classification CNN (**Cnet**) and an image-to-image regression CNN (**Rnet**). Rnet is a simple 4-layer deep CNN that is trained on a carefully designed dataset of palm patches to convert a palm patch directly to its enhanced version. Rnet is able to enhance sufficiently large patches of palmprint (96×96) with adequate robustness. Sufficient depth and an adequate number of kernels enable Rnet to learn complex ridge patterns containing abrupt changes in ridge orientation and frequency. Unlike conventional methods, Rnet does not have to configure its kernels for every patch or perform Fourier analysis. Even in high crease areas, Rnet is able to enhance ridge patterns and subdue creases. Deep

learning solutions are computationally intensive, but by harnessing the power of GPUs, deep learning is quickly replacing classical approaches.

4.3 Palmprint Enhancement Network (*PEN*) Architecture

Figure 4.3 illustrates the overall architecture and flow of *PEN*. Palmprint is first pre-processed to extract only the foreground pixels (palm area) in the image using the DFT-based ROI segmentation procedure proposed in Chapter 3 that effectively removes background noise. Removal of background noise is essential during both offline and online stages in order to limit computation to the valid regions of the image.

After ROI segmentation, palmprint is broken down into patches for subsequent processing. Two separately trained CNNs are used to carry out enhancement. **Cnet**, which is a classification CNN, classifies the palm patch according to dominant local ridge orientation which it predicts directly from the patch. Guided by the orientation prediction of Cnet, palm patches are rotated (if required) to align with kernels of **Rnet**. Rnet, which is an image-to-image regression CNN, then outputs an enhanced version of the patch using pixel-wise predictions. Enhanced palm patches are then rotated back to their original orientation.

In classical methods, orientation and frequency of ridges are estimated pixel-wise, whereas Cnet predicts dominant orientation in a 96×96 patch. A single prediction label for a patch of this size is not a precise estimate of ridge orientations contained within the patch. But *PEN* only needs to predict dominant orientation and feed it to Rnet. Since Rnet has sufficient kernels, depth, and adequate adaptability, it caters to changes in ridge orientation within a patch efficiently. Another important aspect of Rnet is, that it is independent of ridge frequency. Just like changes in orientation, Rnet kernels are able to cater to changes in ridge frequency as well. This adaptability in *PEN* was made possible by designing a training procedure that prepares both Cnet and Rnet for abrupt changes in ridge frequencies and orientations. The training process is explained in the proceeding sections.

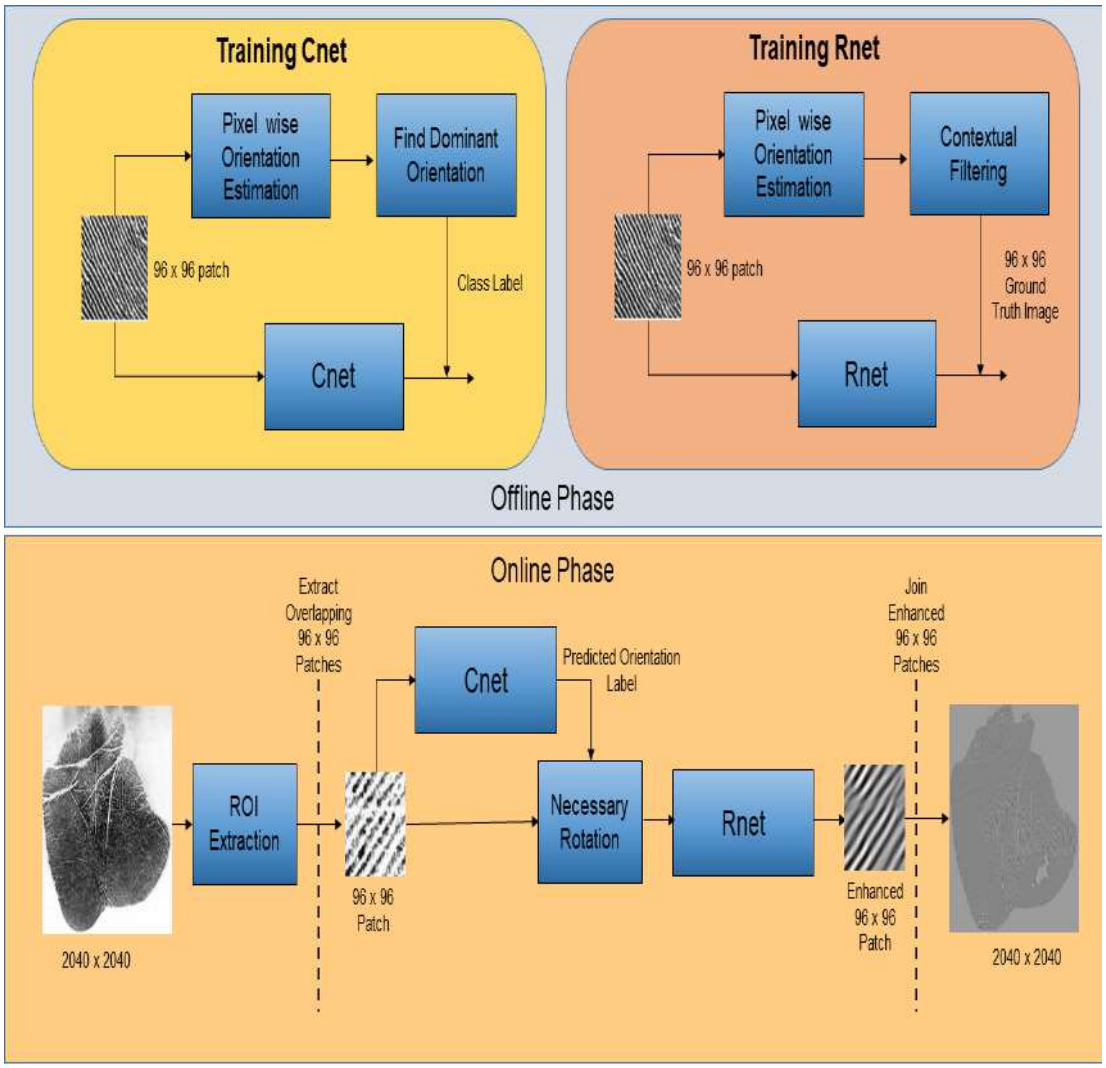

FIGURE 4.3: *PEN* Architecture: Offline phase illustrates the preparation of separate datasets for the training of Cnet and Rnet. The online phase illustrates that after ROI extraction, 96 × 96 patches are extracted from palmprint. Each patch is then passed through Cnet to predict dominant ridge orientation. Based on this information, patches are rotated (if required) and passed through Rnet to yield enhanced palmprint. All patches are later joined to form a complete enhanced palmprint.

A patch size of 96 × 96 is chosen after sufficient experimentation. Prediction scores of Cnet on patches bigger than 96 × 96 were found to be a poor estimate of dominant ridge orientation and patches smaller than these were found to require greater overall computational time. After passing through Cnet and Rnet, all enhanced patches are joined to produce a complete enhanced palmprint. Both CNNs in the offline stage undergo separate training cycles using separate datasets containing palm patches carefully extracted from THUPALMLAB [44] database. Details about *PEN* components are given in subsequent sections.

4.4 Offline Stage

4.4.1 Dataset Preparation for Cnet and Rnet

Training and validation datasets have been prepared using THUPALMLAB high resolution palmprint dataset [44]. THUPALMLAB dataset has been used in all state-of-the-art studies on high resolution palmprints [7, 47, 48, 54, 56–58, 61]. This dataset contains a total of 1280 palmprints from 80 subjects. There are 16 palmprints corresponding to each subject out of which 8 belong to the left palm, and 8 belong to the right palm of the subject. Palmprints in the dataset are of size 2040×2040 taken at $500ppi$. Both Cnet and Rnet are trained on patches of 96×96 pixels. On average one palmprint could produce around 100 to 150 valid patches (of 96×96 pixels) that were found to be suitable for inclusion in training datasets.

The selection of palm patches for training was done carefully keeping the following aspects in mind:

- Since different palm regions (thenar, hypo-thenar, and interdigital) provide different levels of ridge structure quality, special emphasis was paid while creating training data to include patches from all palm regions so that trained CNNs are robust to variable ridge structures.
- Training data included patches with both high and low ridge curvatures
- Training data included patches having different contrast levels of ridge/valley structure
- Training data had a blend of good and poor quality patches to strike a balance between training performance during offline stage and robustness during the online stage.

The above-mentioned considerations were aimed to make *PEN* adaptable to variable ridge quality, ridge curvature, and contrast in the palmprints. The process of associating ground truths and labels with training patches (for Cnet and Rnet)

during the offline stage is explained below.

Using widely used gradient-based methods [63], an orientation image O_{xy} is created that contains pixel-wise ridge orientations in a training patch. O_{xy} is constructed by taking the inverse tangent of the gradient of the palmprint image in vertical G_y and horizontal G_x directions, as given by the equation

$$O_{xy} = 90^{o} + \frac{1}{2} tan^{-1}\left(\frac{2G_{xy}}{G_x - G_y}\right). \tag{4.1}$$

G_x and G_y are calculated using equation 4.2 and equation 4.3, where ∇_x and ∇_y are x and y gradient components of the palmprint image computed using Gaussian filter. G_{xy} is the product of gradients given by equation 4.4.

$$G_x = \sum_{h=-8}^{h=8} \sum_{k=-8}^{k=8} \nabla_x \left(x+h, y+k\right)^2, \tag{4.2}$$

$$G_y = \sum_{h=-8}^{h=8} \sum_{k=-8}^{k=8} \nabla_y \left(x+h, y+k\right)^2, \tag{4.3}$$

$$G_{xy} = \sum_{h=-8}^{h=8} \sum_{k=-8}^{k=8} \nabla_x \left(x+h, y+k\right)^2 \times \nabla_y \left(x+h, y+k\right)^2. \tag{4.4}$$

Since all patches extracted from [44] are not suitable for training purposes, patches of a certain quality were chosen. Variance in ridge orientation within a patch was chosen as the quality measure that determined whether to include a patch in the training set or not. The variance of orientation distribution of a patch with a well-defined ridge/ valley structure is much lower than a patch with broken ridges. The variance threshold was set at 8 which was found to be a good trade-off between achieving good training accuracy and imparting necessary robustness against bad patches which might be experienced in the online stage. Figure 4.4 shows the distribution of pixel-wise orientations available in good and relatively bad patches with a Gaussian curve fitted on them.

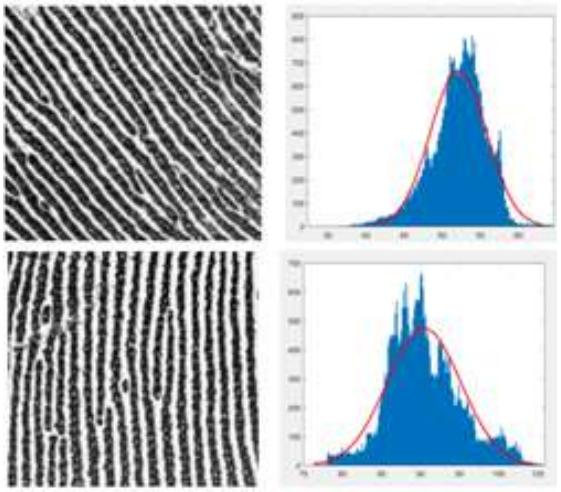

FIGURE 4.4: Ridge orientation distribution: (Top) Uniform, (Bottom) Less Uniform

The mode of the orientation distribution in the patch was taken as an indicator of the dominant orientation as mode gives the most frequently occurring value in data. 0 to 180 degrees range of orientation was quantized in steps of 15 degrees, thereby leaving only 12 classes for patch classification. As a result, a patch with a dominant orientation between 0 to 15 degrees is placed in a bin labeled 15 and a patch with a dominant orientation between 16 and 30 is placed in a bin labeled 30, and so on. Bin labels are used as ground truths or class labels for Cnet.
Once selected patches have been placed in separate bins, gradient-based $2D$ contextual filtering method [48] is used to produce corresponding enhanced versions of these patches. These enhanced versions of patches are used as ground truth for image-to-image learning of Rnet. For the sake of simplicity, Rnet is only trained on patches of a single orientation label. For patches of all other orientations, every patch goes through necessary rotation before passing through Rnet. The process of dataset creation is illustrated in Figure 4.5.

4.5 Cnet - Architecture and Training

Transfer learning was used to fine-tune pre-trained alexnet for the classification of palm patches based on dominant ridge orientation. Alexnet [90] is a deep CNN

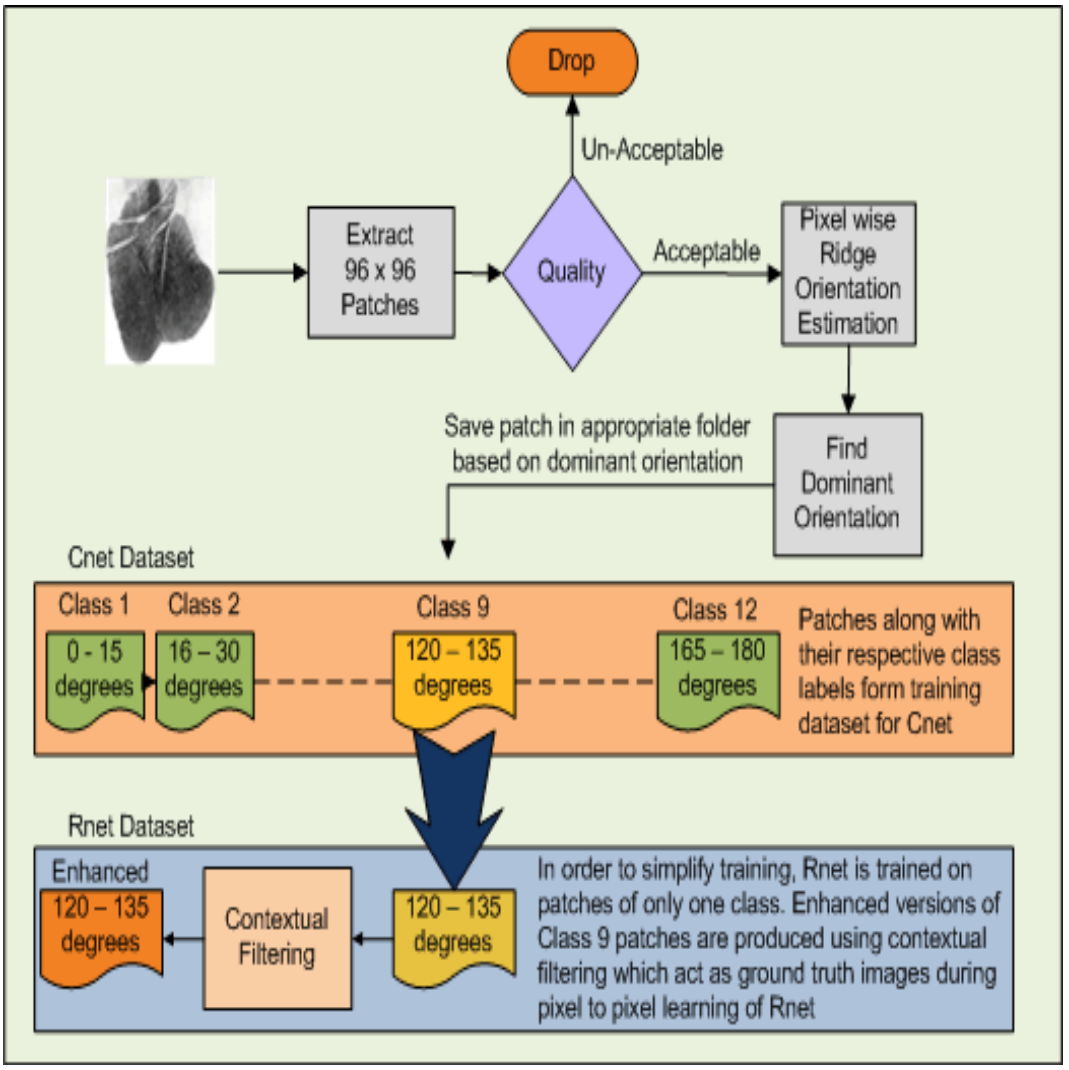

FIGURE 4.5: Training dataset creation process for Cnet and Rnet: Based on the most frequently occurring value in orientation distribution, patches are given class labels and used for training Cnet. Patches from only one class are enhanced using contextual filtering to form ground truth labels for Rnet training

that is trained on a subset of imagenet dataset [91] with over 1 million images. It is originally capable of classifying between 1000 classes. Alexnet consists of 5 convolutional layers, 3 max pooling layers, 2 normalization layers, 2 fully connected layers and 1 softmax layer (Figure 4.6. The last three layers of pre-trained alexnet were replaced with a fully connected layer, a softmax layer, and a classification output layer to classify between 12 distinct ridge orientations. During fine-tuning, the weight learning rate for the new layers was increased to 20 to focus learning more on the new layers while the weight learning rate for the transferred layers was kept to minimal, i.e. 0.0001, to bring minimal changes to these layers. Stochastic Gradient Descent (SGD) training method was used with a batch size of 10.

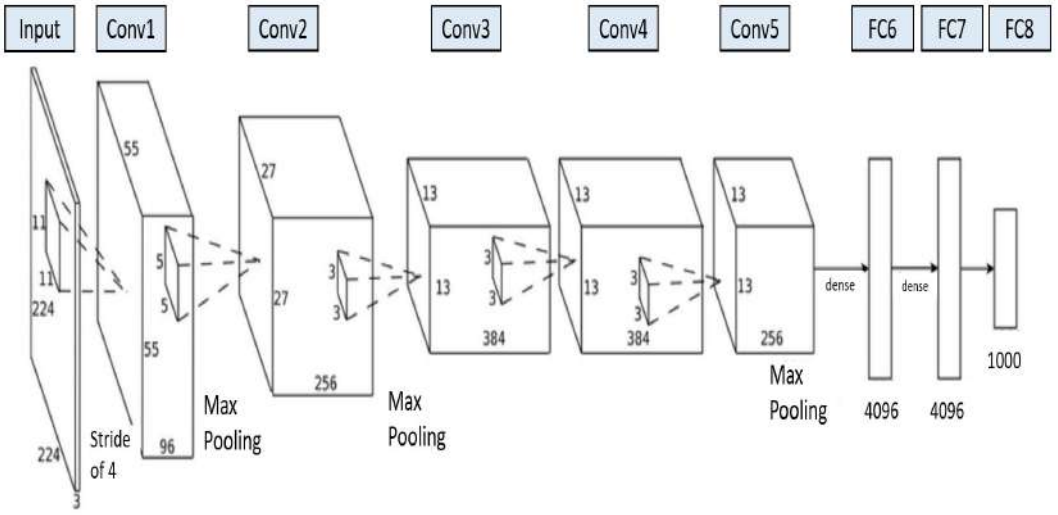

FIGURE 4.6: Alexnet: illustration shows 5 convolutional layers, which are connected to two Fully-connected layers (FC6-7), the output is a fully-connected 1000-way soft-max layer[90]

ReLUs have become the default activation function (equation 4.5) in the recent past because they help train the models faster. Pooling layers help in reducing the dimensions of the image to improve processing speed and also impart robustness in terms of the location of features of interest in the image. Values obtained from the output of the softmax layer (equation 4.6) indicate class predictions.

$$F(x) = max(0, x), \tag{4.5}$$

$$Softmax(x_i) = \frac{\exp(x_i)}{sum_j} \exp(x_j). \tag{4.6}$$

In fingerprints, 5 easily identifiable ridge patterns exist, namely, whorl, left loop, right loop, arc, and tented arc [92] that help in making identification through CNNs easy (Figure 4.7). In palmprint, owing to large ROI, it is not easy to predefine local patterns of ridge structure. Hence, orientation prediction in palms using CNNs is a potentially difficult task. Carefully selected images in the dataset contain a good compromise between good and bad quality images that help in improving training performance while achieving robustness at the same time.
With extensive trials, a prediction accuracy of 90% was achieved for dominant local ridge orientation. Owing to a large number of creases in the palmprints coupled with variable quality of images and lack of predefined patterns in the palmprints,

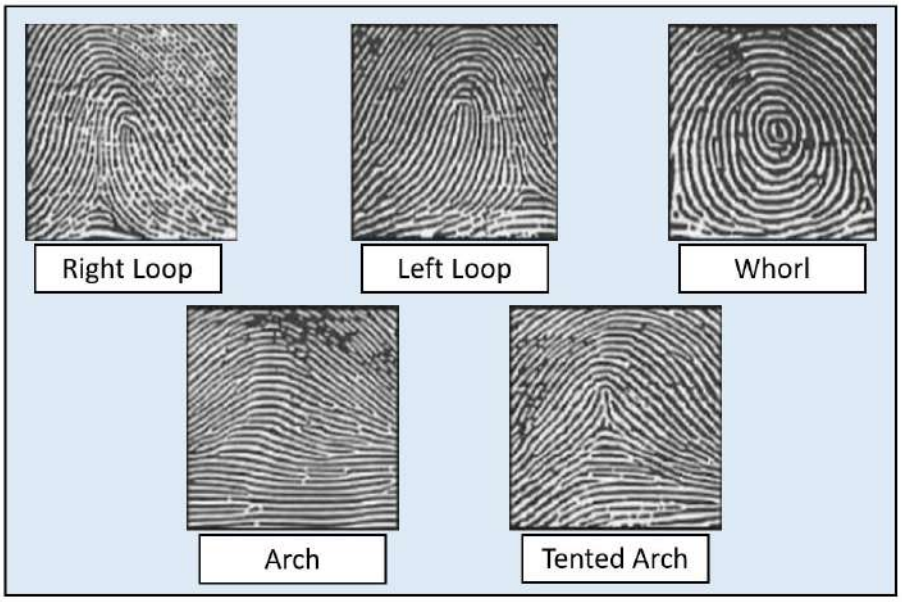

FIGURE 4.7: Ridge patterns in fingerprints [92]

90% is an encouraging accuracy. Inter-class difference of 15 degrees was chosen after sufficient experimentation. Differences lower than 15 degrees produce poor prediction results. 20% of the dataset was used for validation while 80% was used for improving training performance. During online testing, classification errors made by Cnet (if any) were removed using smoothing of class prediction values within neighboring patches.

4.6 Rnet - Architecture and Training

Apart from image classification, CNNs have proved to be extraordinary in image-to-image regression tasks as well. The output layer of a traditional CNN is modified to use it as a regression CNN. Image regression CNNs have attained state-of-the-art performance in computer vision problems such as head pose estimation [93], human pose estimation [94], facial landmark detection [95] or image registration [96]]. Rnet is inspired by recent developments in pixel-to-pixel learning which includes image restoration and image super-resolution [85, 86]. Rnet is a $4-layer$ deep CNN illustrated in Figure 4.8.

Convolutional layers extract features and encode primary components while eliminating unwanted abstractions (creases or background noise) in the image. Pooling

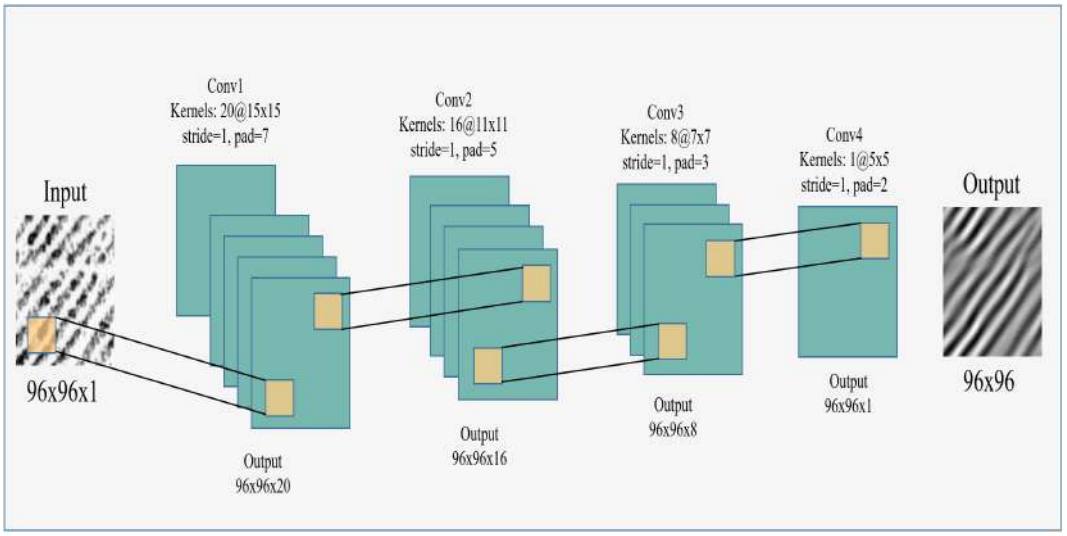

FIGURE 4.8: Rnet architecture: 4 Convolutional layers $Conv1$, $Conv2$, $Conv3$ and $Conv4$ with ReLU activation function. $Conv1$ has 20 kernels of size 15×15, $Conv2$ has 16 kernels of size 11×11, $Conv3$ has 8 kernels of size 7×7 and $Conv4$ has 1 kernel of size 5×5. Euclidean loss (MSE) between the output of $Conv4$ and ground truth patch is used for training

layers not used as the low-level image enhancement problem is more focused on reducing corruptions in low-level features rather than learning complex image abstractions. Secondly, pooling layers remove vital image details while reducing feature space which cannot afford in image enhancement.

Since padded convolutional layers have been used, the size of the output image equals input image i.e., 96×96. Training data for Rnet is in the form (X, Y) where X is a 96×96 patch and Y is the corresponding enhanced version or ground truth for X. The convolution layer can be expressed as:

$$F(X^i) = W_k * X^i + B_k. \tag{4.7}$$

X^i is the *ith* training sample and W_k and B_k represent kernels and biases, respectively. Output A^l of a layer l ($wherel = 1, 2, 3, 4$) is given by:

$$A^l = G^l(W_k^l * X^i + B_k^l). \tag{4.8}$$

G^l is the ReLU activation function of *lth* layer, given by:

$$G^l = max(0, x). \tag{4.9}$$

The output of $Conv4$ (A^4) was used as the final activation. Pixel-wise prediction of enhanced patches given the real patches involves estimating convolutional kernel weights $\boldsymbol{W}$ for all pairs (X^i,Y^i) in a training dataset, where $i = 1, 2, \ldots, N$ and X^i and Y^i represent real and enhanced patch or the ground truth respectively, the objective function is to minimize Mean Squared Error (MSE) expressed in equation (4.10).

$$L(W) = \frac{1}{N} max \sum_{i=1}^{N} ||F(X^i : W) - Y^i||^2. \tag{4.10}$$

Rnet was trained on Caffe [97] with an initial learning rate of 0.00001, momentum of 0.85, batch size of 64 using SGD solver. Figure 4.9 illustrates the training curve of Rnet.

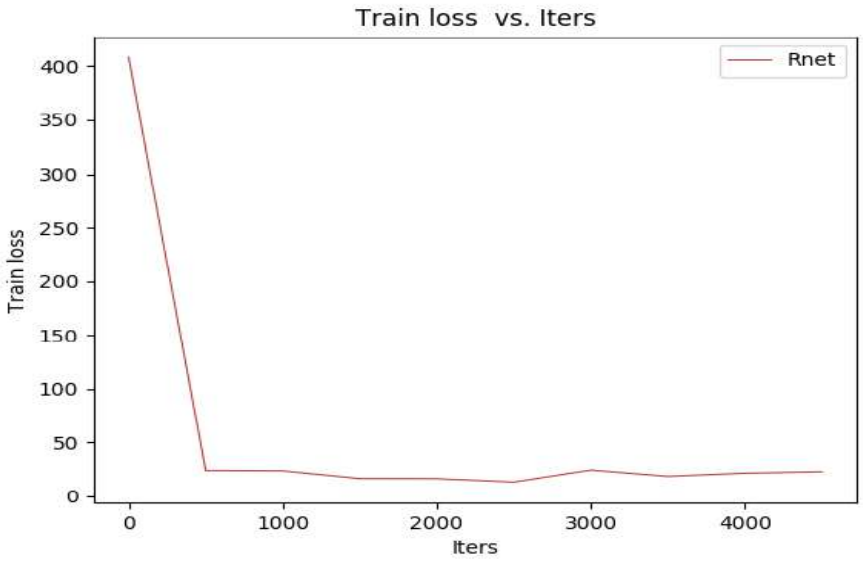

FIGURE 4.9: Training curve of Rnet

4.7 Online Stage

The online stage of the *PEN* is illustrated in Figure 4.10. Palmprint first goes through the ROI extraction to obtain only the foreground pixels. After ROI segmentation, palmprint is broken down into overlapping patches of 96×96. Horizontal and vertical overlapping of 24 pixels is used. Patches are then fed to Cnet for dominant orientation prediction. After orientation prediction, the patch is fed to

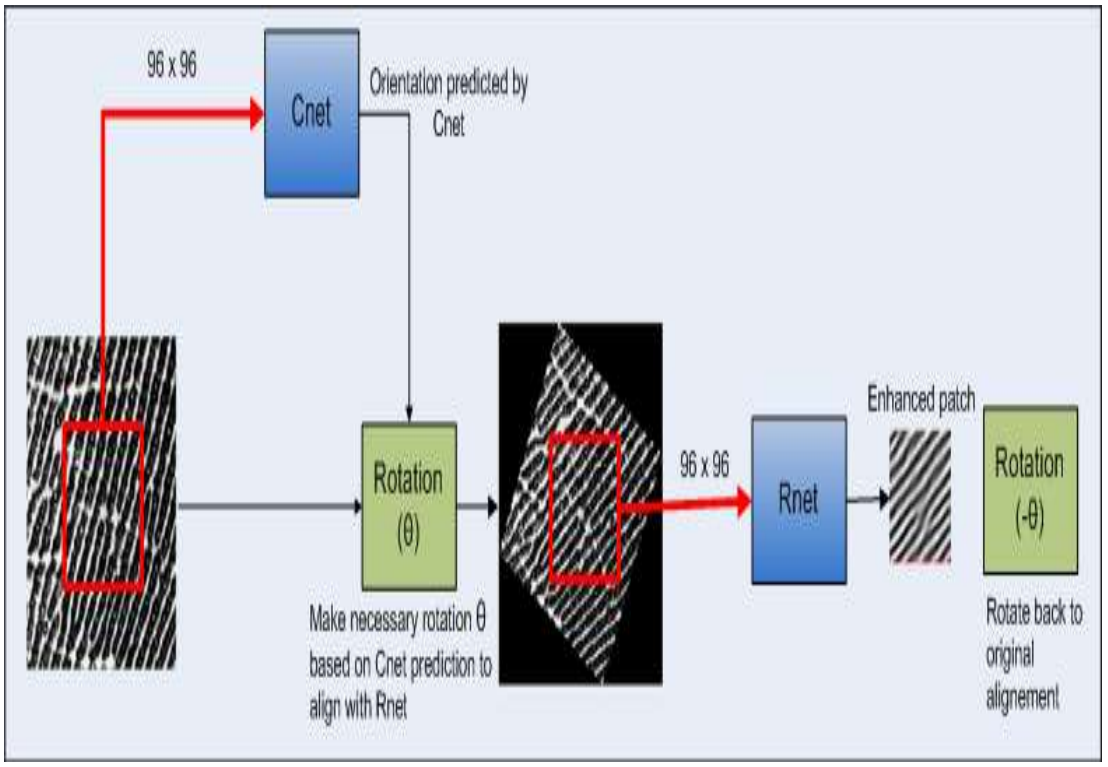

FIGURE 4.10: *PEN*: Online Stage

Rnet. But prior to passing through Rnet, the patch has to go through rotation if required. This is because Rnet has been trained on patches of only one orientation class.

If Cnet predicts the orientation class of a patch to be other than the one Rnet is trained on, the patch undergoes two rotations: one before and one after passing through Rnet. The first rotation will align patch orientation with Rnet and the second rotation will bring the patch back to its original orientation. One alternative to this step is to train Rnet for all orientations which causes Rnet architecture to become unnecessarily complex and training even more difficult. Another alternative is to train separate versions of Rnet for all orientations. It can be easily argued that giving suitable rotation to the patch is a less cumbersome option. At the end, all patches are joined together to output a complete enhanced palmprint. Since overlapping patches are used, discontinuity in patches while joining can be avoided through careful implementation.

Results of palmprint enhancement through *PEN* show the efficacy of the proposed framework. Owing to adaptability imparted during training, Rnet is able to recover ridge structure even in the high-crease areas resulting in the extraction

of reliable features (minutiae). Performance of *PEN* is evaluated from both aspects: recovery of palm ridge structure and identification accuracy of palmprints enhanced by *PEN*. Results of *PEN* are presented in Chapter 5.

4.8 Summary

This chapter proposes a novel deep learning-based two-step Palmprint Enhancement Network *PEN* consisting of two CNNs, i.e., Cnet and Rnet. First of all, classical enhancement methods are described and categorized according to techniques used for the estimation of local ridge orientation, i.e., gradient-based and region-growing techniques. Problems with these methods are highlighted and an out-of-the-box solution is proposed that is able to process considerably large patches of palmprints without having to configure enhancement filters locally. *PEN* converts a palm patch directly to its corresponding enhanced version in the spatial domain. The proposed method removes the need for pixel-wise estimation of ridge orientation and ridge frequency maps.

The architecture of *PEN* is explained with a description of the offline and online stages. The training process for both Cnet and Rnet is explained along with the process of creating training datasets. Careful selection of training patches with the aim of imparting an increased amount of adaptability to *PEN* is described. The architectures of Cnet and Rnet are also explained. In the end, the online stage of *PEN* is presented explaining the overall flow from input palmprint to enhanced palmprint.

Chapter 5

Results of the Proposed Enhancement Method

The proposed Palmprint enhancement network (*PEN*) is designed to eliminate two fundamental assumptions in previous enhancement methods, namely, ridge pattern in a local area is stationary and ridge frequency is constant. In order to validate the performance of *PEN*, it was needed to ascertain the following: 1) *PEN* is robust to abruptly changing ridge orientation and frequency without assuming the ridge pattern to be stationary in a local area 2) Even in the high crease areas, *PEN* is able to extract underlying ridge pattern, and 3) Palmprints enhanced by *PEN* produce promising results during matching.

5.1 Dataset and Experimental Setup

Results are acquired on renowned and challenging THUPALMLAB dataset [44]. Literature review reveals that this high resolution palmprint dataset has been used in all state-of-the-art studies on high resolution palmprints. Dataset consists of 1280 palmprint images taken from both left and right hands of 80 different subjects, with eight impressions per hand. The images have a resolution of 2040 $\times$ 2040 pixels at 500 ppi with 256 grey levels. Performance assessment of *PEN* is based on both, enhancement of palmprints, and accuracy of subsequent matching

or identification. The following metrics are used in evaluating the performance of *PEN*:

- **False Acceptance Rate (FAR)**: Rate of wrongful classification of a non-genuine as genuine
- **False Rejection Rate (FAR)**: Rate of wrongful classification of a genuine sample as non-genuine
- **Equal Error Rate (EER)**: Rate where FAR and FRR are equal

FAR and FRR are interdependent and it is not possible to reduce one rate without reducing the other. That's why the most effective biometric systems are the ones with low FAR and FRR resulting in low EER. Acceptability threshold t in Figure 5.1 can be varied. Selecting a higher threshold means lower FAR but higher FRR and vice versa.

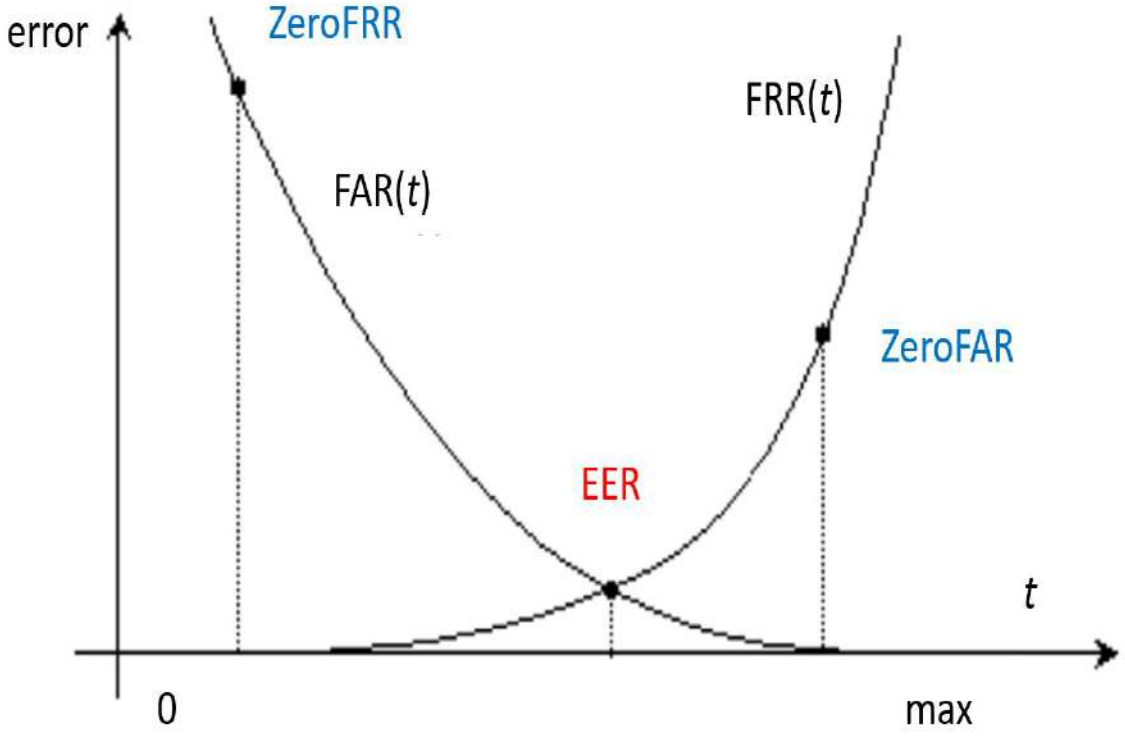

FIGURE 5.1: EER denotes when FAR=FRR, ZeroFAR denotes FRR when FAR=0, ZeroFRR denotes FAR when FRR=0, t is acceptability threshold

Apart from EER, **Rank-1 accuracy** and Detection Error Trade-off (**DET**) results

are also calculated to verify the quality of enhancement. Results are compared with the state-of-the-art. In addition to the results, section 7.3 contains details about architectural choices for Rnet and analysis of the chosen architecture, information about the combination of hyperparameters, and the impact of Cnet on the performance of Rnet.

All experiments pertaining to this article are performed on a system equipped with Intel i7-6700 HQ CPU, 8 GB of RAM, and an NVIDIA GTX 960M GPU with 4GB memory.

5.2 Enhancement Results

Enhancement results of *PEN* are compared with contextual filtering methods [47, 48, 50–55], since they have been most popularly used for recovering palmprint ridge structure. Figure 5.2 shows a variety of regions extracted from different palmprints. Performance of *PEN* can be observed to be robust to the quality of input patches. Figure 5.3, shows the conversion of a complete palmprint to its enhanced version. Figures 5.2 and 5.3 highlight the adaptability of *PEN* on high curvature areas of palmprints which are challenging for any enhancement algorithm. It can be seen in Figure 5.3, that the hypothenar region provides the best results for enhancement and subsequent minutiae extraction due to smooth ridge orientation and less number of creases. Comparison between *PEN* and state-of-the-art contextual filtering methods is drawn in Figure 5.4. Some poor-quality patches have been chosen for comparison.

It can be seen that *PEN* performs better at maintaining ridge consistency than contextual filtering methods. This is because, in classical methods, the results of contextual filtering depend mainly on the estimation of underlying ridge orientation. In high crease areas, both gradient-based and region-growing (Chapter 4) methods of orientation field estimation end up detecting creases rather than ridge lines. This results in the configuration of contextual filters according to crease orientation rather than ridge lines. Consequently, contextual filters tend to enhance creases instead of ridge lines in high crease areas.

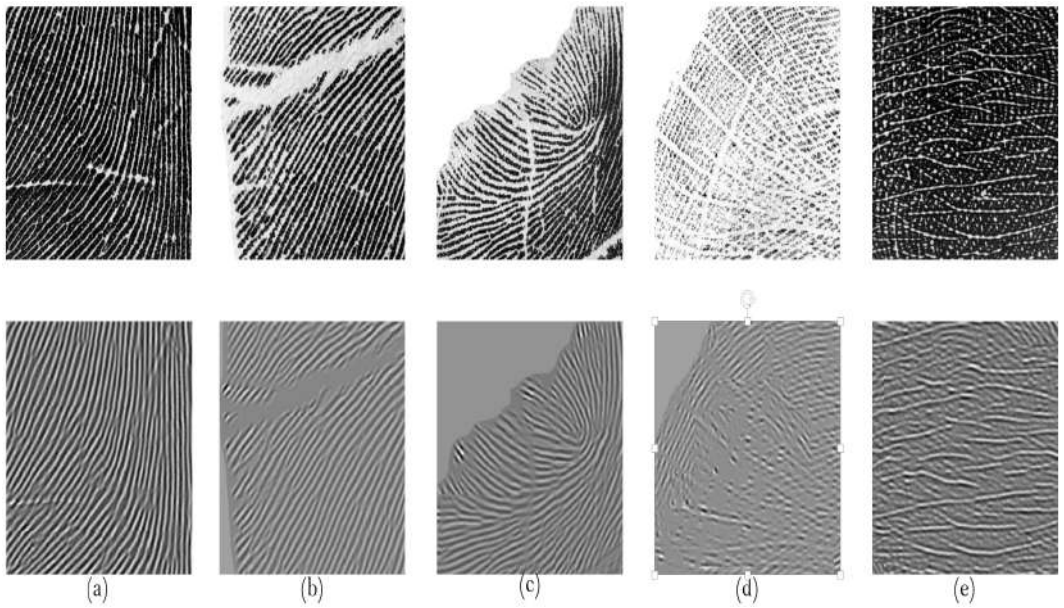

FIGURE 5.2: Enhancement Results of *PEN*. (a) Steady ridges in the hypothenar region (b) Region with major crease, (c) Region with high curvature and inconsistent ridge frequency, (d) Region with poor contrast and broken ridges, (e) Non-existent ridge structure with frequent creases

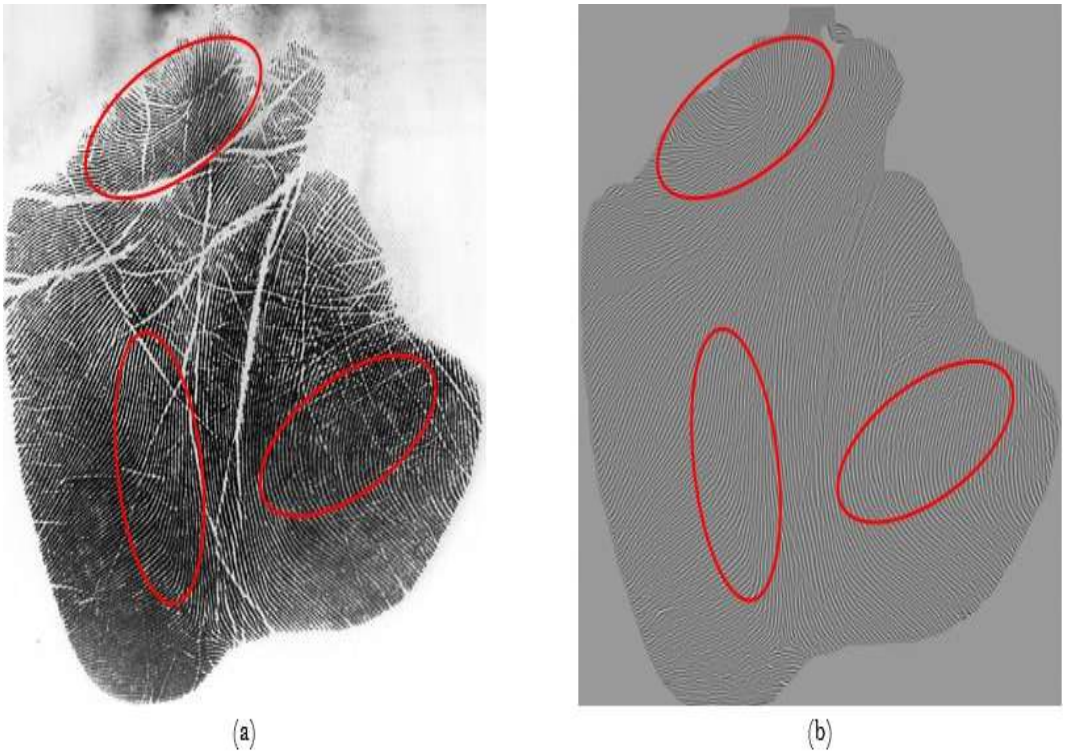

FIGURE 5.3: *PEN* on high curvature areas with inconsistent ridge frequency indicated in red. (a) Original Palmprint (b) Enhanced Palmprint

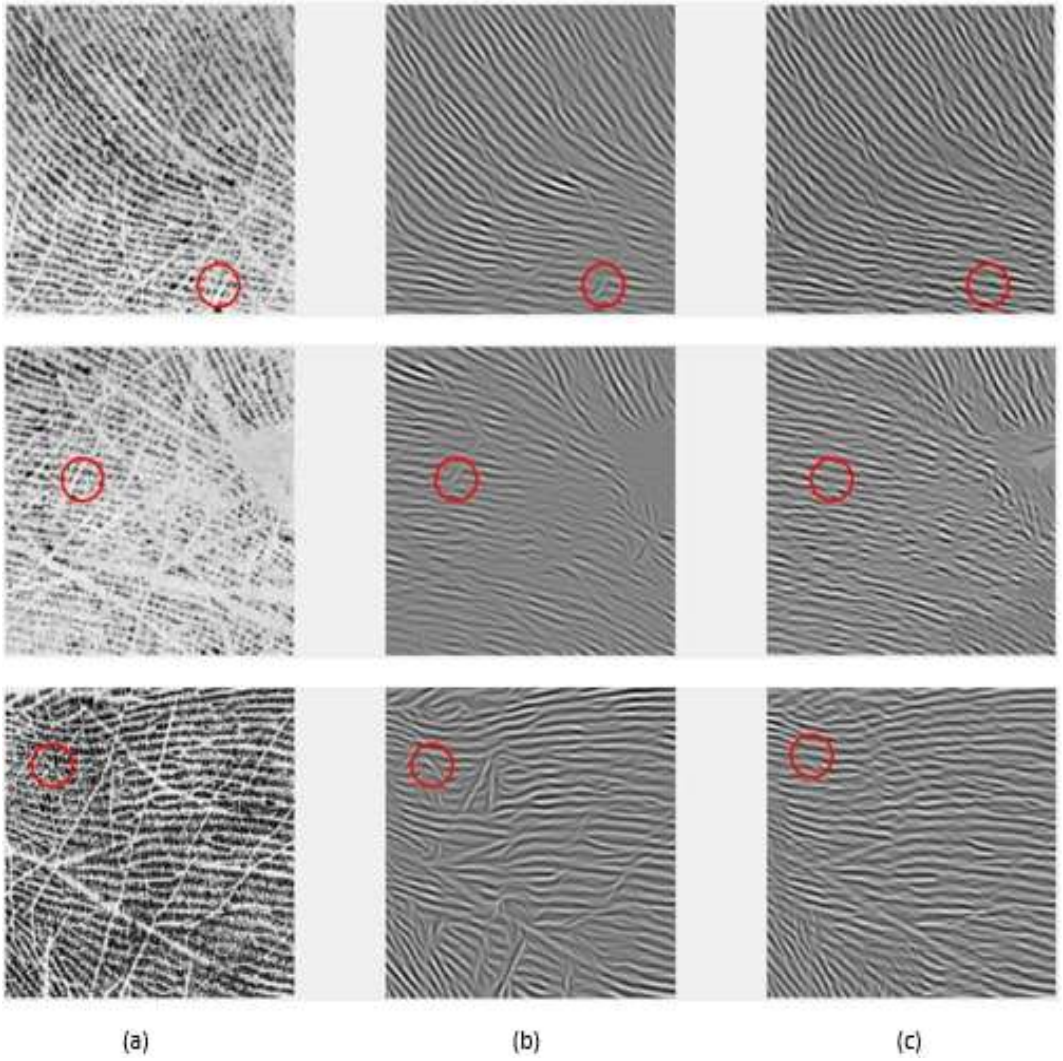

FIGURE 5.4: Enhancement Comparison between *PEN* and contextual filtering. (a) Original palm patch (b) Enhancement by contextual filtering (c) Enhancement by *PEN*. Results show that contextual filters applied in small local areas containing creases (indicated by red circles) pick up contextual information pertaining to creases and end up enhancing creases instead of ridges, while *PEN* is able to enhance ridge structure and subdue creases

5.3 Matching Results

In order to further verify the enhancement performance of *PEN*, the identification accuracy of palmprints enhanced by *PEN* is also evaluated. Minutiae-based palmprint matching scheme was used where minutiae are described as the ridge endings or ridge bifurcations (Figure 1.12). A recently published two-stage minutiae-based

matching technique was used [48], with the difference that instead of using contextual filtering-based enhancement of palmprints, *PEN* was used. Without paying much focus to post enhancement and matching algorithms, *PEN* were able to achieve remarkable matching results which strengthens confidence in PEN. Minutiae are extracted from the enhanced palmprint produced by *PEN* after binarization and thinning [55]. Spurious or false minutiae are removed in the post-processing.

Valid minutiae extracted from each palmprint are then represented as $Minlist = [m_1, m_2....m_N]$. Each minutiae m_j is a triplet, $m_j = [x_j, y_j, \theta_j]$, where $j = 1, 2....N$. (x, y) are the coordinates of minutiae and θ is the ridge orientation at the minutia point with respect to the x-axis. Each minutia is then encoded based on its n nearest neighboring minutiae. The value of n is chosen to be 10. Encoding of all minutiae in a palmprint is stored as its $PalmCode$. During matching, $PalmCode$ of input palmprint called the query palmprint is matched with every other $PalmCode$ in the database. Matching occurs in two stages; local and global. At the local stage, each out of M minutiae in the query palmprint is matched with each out of N minutiae in the candidate palmprint. For a single minutia pair to match, at least th_n amount of neighboring minutiae need to match. Value of th_n is chosen to be 5, i.e., at least 5 out of 10 neighbors of both minutiae must match. The similarity score is calculated for every minutia in the same way. Minutiae matched at this local stage are further processed at the global stage, while unmatched minutiae are discarded. At the global stage, each minutia is encoded again with respect to its top 20 nearest neighbors. The similarity score of these minutiae is calculated in the same way as on the local stage to give the final matching score of palmprints.

Equal Error Rate (**EER**) is used to assess the accuracy of matching results. EER is calculated by computing the False Acceptance Rate (**FAR**) and the False Rejection Rate (**FRR**). EER represents a threshold where both FAR and FRR are equal. FRR (also called genuine matching) is determined by comparing each palmprint sample of a hand of a subject with other samples of the same subject's hand. The total number of genuine matches is calculated as $((87)/2)802 = 4480$. FAR (also called impostor matching) is determined by comparing the first palmprint

sample of each hand of a subject with the first palmprint sample of the same hand of the remaining subjects. The total number of imposter matches is calculated as $(8079)/2)2 = 6320$.

Results of the matching algorithm on palmprints enhanced by *PEN* are shown in Table 5.1. Without spending any effort on optimization of minutiae encoding and matching algorithm for deep learning-based enhancement, *PEN* was able to achieve EER of 0.15 (see *PEN* in Table 5.1) which is remarkably low. Matching scores of palmprints were inspected and it was found that palmprints performing poorly in the matching stage showed some common properties. Three properties of palmprints were picked to identify poorly performing images, namely, the number of valid minutiae ($MinValid$) in a palmprint, mean ridge curvature ($CurvMean$) of complete palmprint, and mean of local curvatures around each minutia in a palmprint in a 15×15 window ($LocCurvMean$). The average of these three attributes is calculated for each palmprint to depict palmprint quality, i.e.,

$$PalmQuality = Avg(MinValid, CurvMean, LocCurvMean). \tag{5.1}$$

$PalmQuality$ was calculated for each palmprint and the results were analyzed. In proposed implementation, the value of $PalmQuality$ varied from 0.40 to 1.32. It was observed that palmprints that severely affected EER results had produced the lowest $PalmQuality$ values. During the matching stage, the value of th_n was lowered to 4 instead of 5 for palmprints that produced $PalmQuality$ values lower than 0.45. Incorporating this simple adaptability into the system lowered the overall EER to 0.13 (see PEN_{adapt} in Table 5.1) which is lower than most state-of-the-art methods. EER curve is illustrated in Figure 5.5. This value of EER is impressive because it comes without making any improvement to the minutiae encoding and matching algorithm. And by only dropping 10 palmprints with the lowest $PalmQuality$ from the matching stage as a pre-processing step, EER was calculated as 0.06 (see $PEN_{preproc}$ in Table 5.1).

Seeing EER results, it can be easily argued that by using superior minutiae encoding and matching techniques [58] and limiting feature search space to only

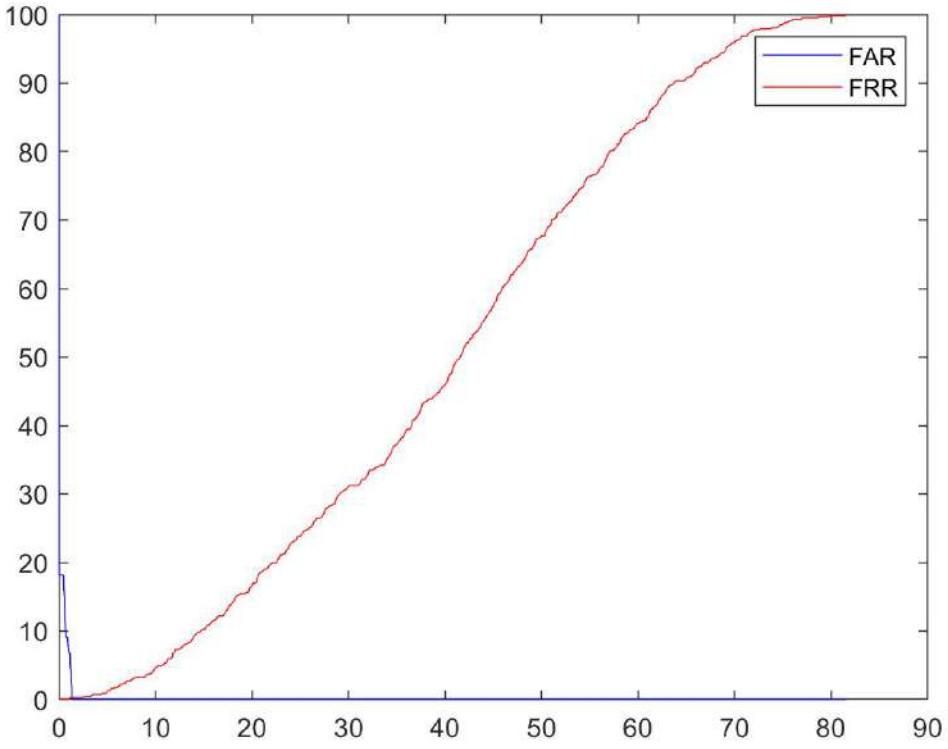

FIGURE 5.5: Equal Error Rate (EER) of palmprints enhanced through *PEN*

good quality areas using pre-processing techniques like [15], proposed enhancement scheme can provide the foundation for the best minutiae matching results. In order to further verify results, FRR values at different FAR thresholds were calculated and compared with other state-of-the-art methods. Results are presented in Table 5.2 which shows that the proposed method performs favorably in comparison to state-of-the-art.

The detection error trade-off (DET) graph is an alternative metric to EER. DET graph plots the False Rejection Rate (FRR) against the False Acceptance Rate (FAR) on a logarithmic scale. DET graph showing a comparison between PEN and other enhancement schemes that use similar minutiae encoding and matching algorithms is presented in Figure 5.6. The graph shows that PEN gives lower FRR values as compared to other enhancement schemes. This is a testament to the quality and reliability of features extracted using the proposed enhancement scheme.

As another test of matching accuracy, the rank-1 identification rate was calculated

Method	**Ridge orientation estimation**	**Enhancement**	**EER**
Jain et al.[7]	Region-growing	Gabor Filters	5.04
Dai and Zhou[56]	Region-growing	Gabor Filters	2.99
Tariq et. al.[47]	Gradient-based	Gabor Filters	0.38
Ghafoor et. al.[48]	Gradient-based	Gabor Filters	0.18
Hussain et. al.[58]	Region-growing	Gabor Filters	0.04
Khodadoust et. al.[14]	Curved Gabor Filters	Directional Filters	2.01
PEN	Cnet	Rnet	0.15
PEN_{adapt}	Cnet	Rnet	0.13
$PEN_{preproc}$	Cnet	Rnet	0.06

TABLE 5.1: EER Comparison of *PEN* with state of the art

Method	**FRR at FAR < 0.01%**	**FRR at FAR < 0.001%**	**FRR at (%) FAR = 0**
Jain et al. [7]	17.32%	19.43%	22.5%
Dai and Zhou [56]	8.78%	10.45%	11.58%
Tariq et. al. [47]	>0.38%	>0.38%	>0.38%
Hussain et. al. [58]	0.12%	0.14%	0.24%
PEN	0.20%	0.25%	0.44%

TABLE 5.2: EER Comparison of *PEN* with state of the art

and compared with the state-of-the-art. In order to calculate the rank-1 identification rate, each palmprint is compared with at least two palmprints of each hand of all 80 subjects in the THUPALMLAB database. It can be seen in Table 5.3, *PEN* helps in attaining good rank-1 accuracy.

Method	**Rank-1 Identification Rate**
Jain et al. [7]	82.0
Dai and Zhou [56]	91.7
Liu and Feng [13]	99.3
PEN	99.6

TABLE 5.3: Rank-1 identification rate comparison

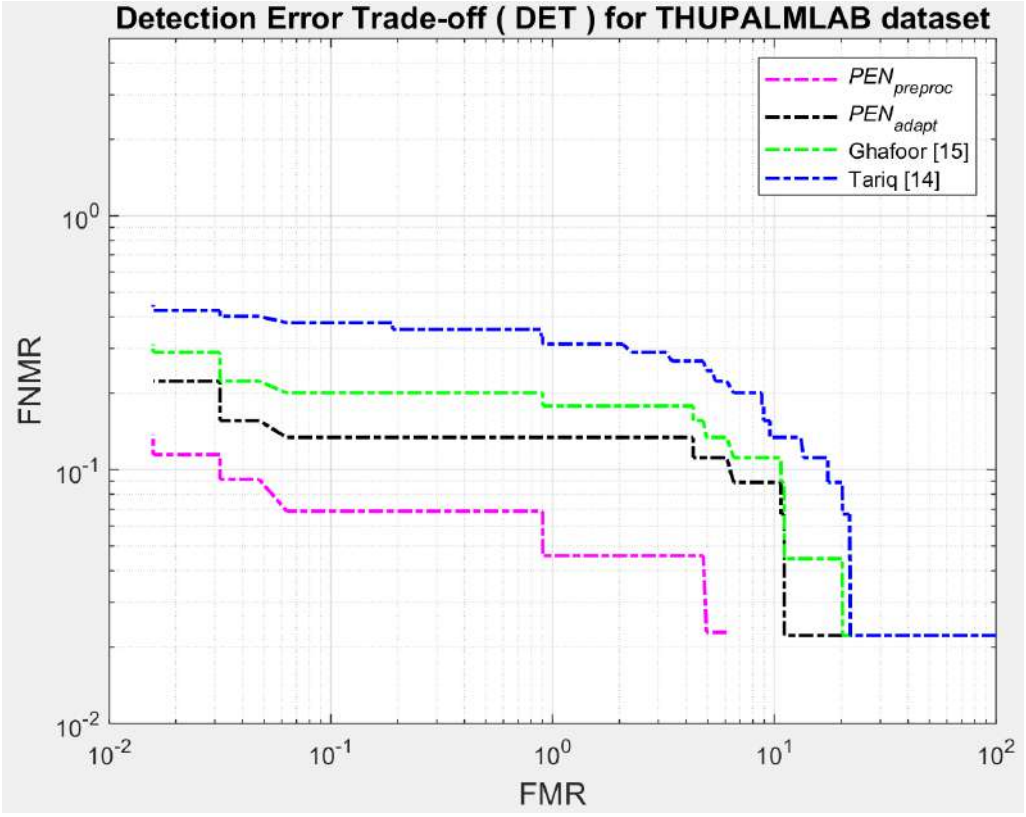

FIGURE 5.6: DET graph: comparison with different enhancement schemes using similar minutiae encoding and matching method

5.4 Discussion

5.4.1 Analysis of Rnet Architecture

Traditionally in CNNs, the number of kernels increases in deeper layers, and kernel size decreases. Initial layers act like Gabor filters and extract low-level features such as edges, blobs, etc. which can be represented using a small number of kernels. On the other hand, deeper layers are trained to extract various high-level features which need a large number of kernels for correct representation. But for low-level image enhancement problems, this proves counter-productive. It was observed that a large number of kernels in deeper layers make Rnet more receptive to complex structures in palmprint which include creases and other kinds of noises. This is equivalent to enhancing noise rather than the original ridge pattern and this is exactly what should be avoided. Hence, the number of kernels in Rnet is gradually decreased in every subsequent layer to keep the focus on low-level

features only, i.e., ridge lines. Further details about Rnet architecture are listed in Table 5.4.

Layer	**Output Shape**	**Trainable Parameters**	**Convolution (%) layer details**
Input	96x96x1	0	-
Conv1	96x96x20	4520	Kernels: 20@15x15, Stride=1, Pad=7
ReLU	96x96x20	0	-
Conv2	96x96x16	38736	Kernels: 16@11x11, Stride=1, Pad=5
ReLU	96x96x16	0	-
Conv3	96x96x8	6280	Kernels: 8@7x7, Stride=1, Pad=3
ReLU	96x96x8	0	-
Conv4	96x96x1	201	Kernels: 1@5x5, Stride=1, Pad=2

TABLE 5.4: Rnet architectural details

Extensive tests were performed to optimize the performance of Rnet. Various alterations of the following hyper parameters were performed to arrive at the best results:

- **Number of convolutional layers**: The number of convolutional layers was changed from 3 to 6. Architectures with convolutional layers between 4 to 5 showed promising results.
- **Size of kernels**: Different sizes of kernels were employed in different layers to improve enhancement results.
- **Training optimization algorithms**: Training was conducted using SGD [98], [99] and Adam [100] optimization algorithms.
- **Activation layers**: Performance comparison of ReLU and Leaky-ReLU was performed.
- **Pooling Layers**: Pooling layers are not used as they reduce feature space which is not desirable in pixel-to-pixel image learning problems.

During experiments, different variations of Rnet architecture were tested to find the best results. The architecture illustrated in Figure 4.8 was empirically chosen to be the best performing architecture after extensive testing. Details of tested Rnet architectures and matching results are given in Table 5.5 along with the number of convolutional layers, the number and size of kernels, and trainable parameters (paras). EER results in Table 5.5 show that increasing the number of convolutional layers and kernels deteriorates EER. This is because palmprint enhancement is a low-level image enhancement problem. Deeper and more complex architectures end up extracting high-level objects in the image such as creases and noise which undermines the ridge pattern in palmprint. Keeping convolutional layers up to 4 and gradually reducing the number and size of kernels in the 2nd and 3rd layers gives the best results.

Rnet Variations	**Conv layers**	**Conv kernels**	**Paras**	**EER %**
Rnet	4	20@15x15, 16@11x11, 8@7x7, 1@5x5	49,737	0.15
Rnet-var1	4	20@11x11, 16@11x11, 8@11x11, 1@11x11	57,641	0.51
Rnet-var2	4	32@13x13, 16@11x11, 12@9x9, 1@7x7	58,289	0.33
Rnet-var3	4	20@15x15, 16@9x9, 8@7x7, 1@5x5	36,937	0.36
Rnet-var4	5	20@15x15,16@11x11,8@7x7,4@7x7,1@5x5	51,209	0.26

TABLE 5.5: Different variations of Rnet architecture

Figure 5.7 shows training curves of Rnet and its variations. Although variations of Rnet show a similar training curve as Rnet, their performance during the matching stage was found to be below par. This is due to the fact that their kernel sizes and depth of convolutional layers were not suited for the low level ridge enhancement tasks. Enhancement using these variations created some artefacts during post processing which were falsely classified as false minutiae which deteriorates matching accuracy. Recovery of palm ridges is aimed at removing discontinuities in the ridge structure found in real-world palmprints. For that enhancement techniques need to smooth the ridge structure where discontinuities occur.

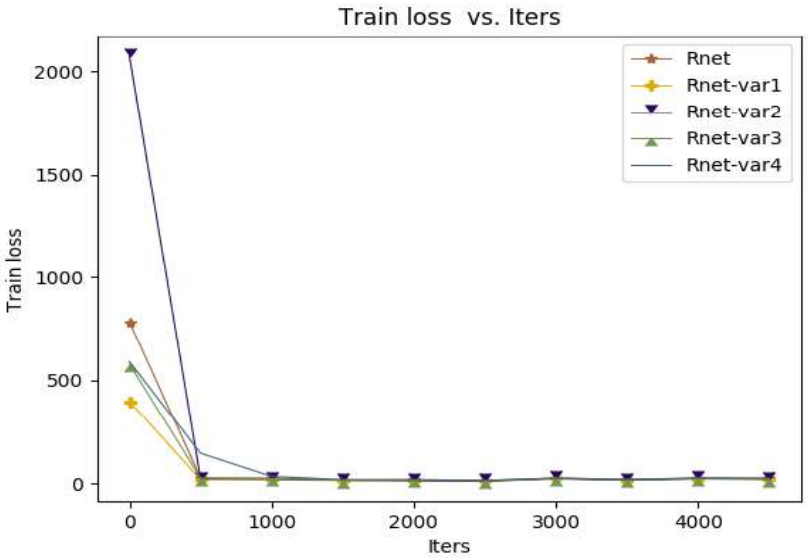

FIGURE 5.7: Training curves of Rnet and its variations

While comparing the performance of various training optimization algorithms, SGD was found to be most suited to the problem of palmprint enhancement. Comparison of SGD, Adam, and RMSProp optimization algorithms during training is illustrated in Figure 5.8.

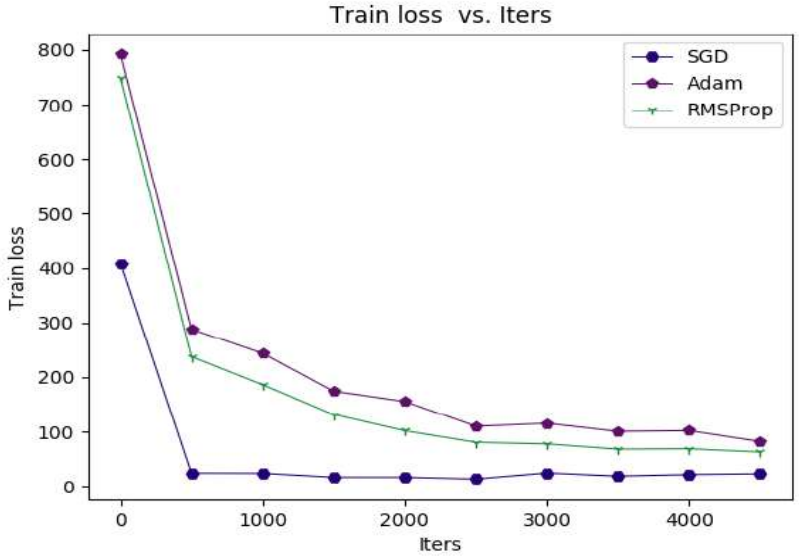

FIGURE 5.8: Comparison of SGD [98], RMSProp [99] and Adam [100] optimization algorithms during Rnet training

5.4.2 Comparison of Proposed Rnet with Recent Deep Learning Paradigms

In the recent past, CNNs have seen remarkable success in image-to-image regression. Various image processing fields, especially image restoration, image denoising, and image segmentation [85–89] have seen an extensive application of image-to-image regression. Deep encoding-decoding architectures consisting of symmetric convolutional-deconvolutional layers have been a popular choice for calculating end-to-end mappings between corrupted and clean images. U-net [89] is an advanced type of encoder-decoder architecture containing additional connections between encoder and decoder parts. It was first proposed for medical image segmentation but gained an overwhelming success in other image processing problems too. Residual network (ResNet) [84] is another remarkably successful deep learning framework. ResNet has mitigated the problem of training very deep networks by using residual blocks. Main concept of a residual block is that output of a layer is not only fed to next layer but also added to the output of another layer much deeper in the path. This connection to the deeper layer is called a "skip connection". These connections also help gradients to flow without interruptions during training, thus overcoming the problem of vanishing gradients in deeper layers.

Although these architectures are immensely popular but to the best of our knowledge, they have not been used for the specific problem of palmprint enhancement. In this section, we compare palmprint enhancement performance of the proposed Rnet architecture with U-net and ResNet. U-net is originally designed for image segmentation and has been adapted for image-to-image regression. For ResNet, we have used ResNet-18 with the necessary modifications for it to perform image-to-image regression as it is originally a deep image classification network.

Figure 5.9 illustrates a thinned palm patch derived from a palm patch enhanced by Rnet, U-net, and ResNet. It can be seen that Rnet is able to maintain the continuity of ridges while very deep U-net and ResNet models create some artifacts during enhancement that are manifested as anomalies in the thinned images. These anomalies are depicted by red circles in Fig. 17. These anomalies act as

false minutiae and deteriorate matching accuracy. This is because deeper layers and a large number of convolutional kernels in very deep CNN architectures tend to learn complex structures in the image and end up creating false structures in the enhanced image.

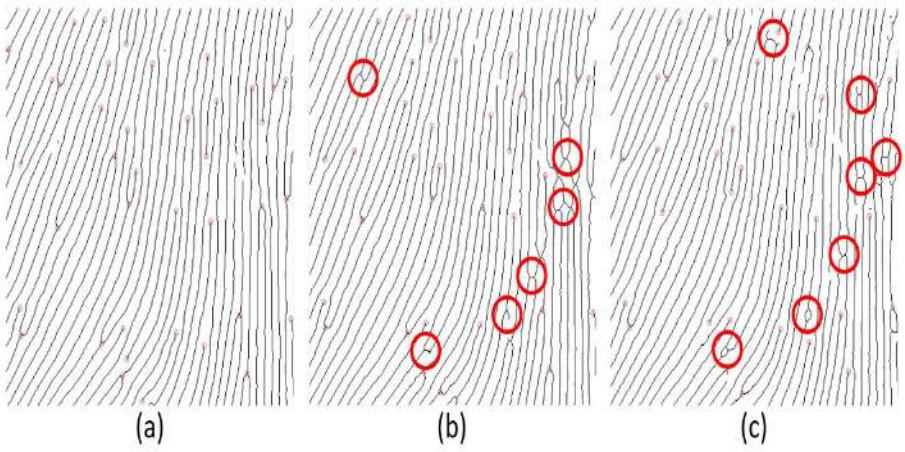

FIGURE 5.9: Palm patch after thinning: (a) Rnet, (b) U-net, (c) ResNet

Table 5.6 presents two more comparisons between the proposed Rnet, U-net, and ResNet. Firstly, it compares the number of trainable parameters in all three networks. Training time and training complexity are directly linked to the number of trainable parameters in a network. Secondly, Table 5.6 presents the comparison of matching accuracy achieved by the features extracted from all three networks. It can be seen that Rnet performs favourably in comparison to the other two. Features extracted after enhancement by Rnet show superior matching accuracy. Also, due to a very simple architecture, Rnet has very little number of trainable parameters that require very little training time and offers minimal complexity as compared to U-net and ResNet.

Model	**Trainable Parameters**	**EER(%)**
Rnet	49,737	0.15
U-net [89]	31 Million (approx)	0.78
Resnet18 [84]	11 Million (approx)	1.41

TABLE 5.6: Comparison between the proposed Rnet, U-net [89] and ResNet [84]

Both Figure 5.9 and Table 5.6 reiterate the fact that deeper convolutional networks, although very successful for complex image mappings, are not suited for low level palmprint enhancement problems. As a further proof of this concept, proposed Rnet architecture consisting of 4 convolutional layers is compared with Rnet-var4 (section 5.4.1) consisting of 5 convolutional layers, and results are illustrated in Figure 5.10. It can be seen, that increasing number of convolutional layers deteriorates ridge continuity which is depicted by red circles in Figure 5.10(b).

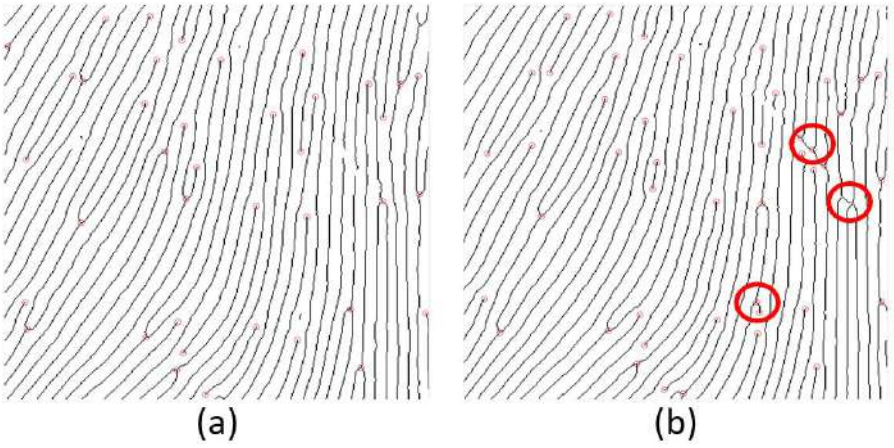

FIGURE 5.10: Comparison of proposed Rnet with Rnet-var4: (a) Rnet: 4 layers, (b) Rnet-var4: 5 layers

5.4.3 Impact of Cnet on Performance of Rnet

In order to simplify design and training, Rnet is trained on palm patches of a specific ridge orientation. Patches with other orientations are rotated to align with Rnet and then passed through it. This requires the estimation of dominant ridge orientation in a patch. During experiments, the gradient-based method was used at each point (x, y) to provide ridge orientation in a patch of 96 $\times$ 96. Since ridge orientation is not constant in the patch, dominant orientation was estimated using two statistical measures separately, namely, mean and mode. However, after extensive trials, it was ascertained that the estimation of dominant ridge orientation using these statistical properties did not give optimal results. As a result, trials were conducted to predict dominant orientation in patches using a classification CNN (Cnet). It was found that matching results of palmprints enhanced by Rnet were better when it was aided by Cnet. EER results with and

without Cnet are given in Table 5.7.

Dominant Orientation Estimate	EER%
Mean	0.71%
Mode	0.24%
Cnet	0.15%

TABLE 5.7: EER comparison of *PEN* with and without Cnet

The process of training dataset creation and training is explained in section 4.4. Various architectures were trained using transfer learning to find the best results. The results of tests are presented in Table 5.8. Owing to superior accuracy and relatively lower prediction time, alexnet was chosen as the best candidate for Cnet in *PEN*.

Architecture	Accuracy%	Average Prediction Time (secs)
Googlenet [101]	88	0.017
Alexnet [90]	90	0.013
Resnet18 [84]	87	0.016
Resnet50 [84]	86	0.022
Squeezenet [102]	86	0.010

TABLE 5.8: Comparison of candidates for Cnet in terms of accuracy and prediction time (one patch of 96x96)

5.5 Computational Complexity of Rnet

Deep learning solutions have high computational requirements. The recent success of deep learning owes a great deal to the use of graphical processing units (GPUs). By harnessing the power of GPUs, large blocks of data can be processed in parallel. Image processing problems are pixel-based where the same operation is performed on all pixels and the values of pixels do not depend on each other. These operations are more suited to parallelism offered by GPUs rather than the sequential processing of CPUs. A comparison of computation requirements is drawn between

Rnet and Gabor filter-based enhancement [50] below.

The computational complexity of both approaches is assessed by evaluating the number of multiplication-accumulation (MAC) operations. For an input image of size $M \times M$, using a Gabor filter of size $m \times m$, expressions (5.2) and (5.3) give multiplication and addition operations for Gabor filter-based enhancement respectively, i.e., :

$$M^2m^2 + 4M^2(1 + \frac{1}{m}(4m + 0.5log_2(m) + 1)), \quad and \tag{5.2}$$

$$M^2m^2 + 2M^2(1 + \frac{2}{m}(3m + log_2(m) + 2)) \tag{5.3}$$

Whereas MAC operations of CNNs (single convolutional layer) are calculated as

$$K^2 \times C_{in} \times H_{out} \times W_{out} \times C_{out}. \tag{5.4}$$

Where K is the size of a square convolutional kernel, H_{out} and W_{out} are the height and width of output which remain constant in Rnet. C_{in} and C_{out} are the input and output channels of a single layer, respectively. In the case of Rnet, MAC operations are calculated for each of the 4 convolutional layers using 5.4 and added up. This gives us total MAC operations in the forward pass of Rnet. Analysis of the computational complexity of both Gabor-based enhancement and Rnet-based enhancement is presented in Table 5. Comparison is drawn on the input image size of 96×96. Value of m for Gabor filters in (5.2) and (5.3) is chosen to be 17. It can be seen that Rnet-based enhancement has much more computational overhead for a CPU, but by using the parallelism offered by even a modest GPU this predicament can be easily overcome.

Enhancement	**Approx MAC Operations (Mns)**	**Average Time (seconds)**	
		CPU	**GPU**
Gabor filters	5.7	0.15	-
Rnet	456	0.033	0.01

TABLE 5.9: Comparison: Gabor-based and Rnet-based enhancement

5.6 Summary

This chapter presents multiple facets of the performance of *PEN*. It first compares the quality of enhancement to state-of-the-art methods by illustrating the recovery of ridges in high-curvature and high-crease areas. Subsequently, enhancement results are corroborated by presenting the palmprint identification accuracy on the features extracted through *PEN*. Matching performance on multiple metrics is presented, i.e. EER, DET graph, and rank-1 identification. At the end, Architectural choices for Cnet and Rnet during the design of *PEN* are discussed.

Chapter 6

Proposed Minutiae Selection Algorithm *(MSA)* For Computationally Efficient Palmprint Matching

6.1 Background

Section 2.2.8 briefly highlights the presence of spurious or false minutiae in the palmprints. These minutiae exist because of discontinuities in the ridge structure of the palm. The main contributors to these discontinuities are minor and major creases in the palmprint. These discontinuities in the ridge lines are falsely classified as minutiae by feature extraction algorithms. Another place where false minutiae appear is the boundary of the palmprints. Ridge endings at the palm boundaries are also wrongly classified as minutiae. Removal of spurious minutiae is essential for two reasons: Firstly, the presence of spurious minutiae artificially increases computational overhead during matching, and secondly, spurious minutiae deteriorate the matching accuracy of the matching algorithm. False or spurious minutiae can be identified in the thinned image and can be classified into six different types (Figure 6.1).

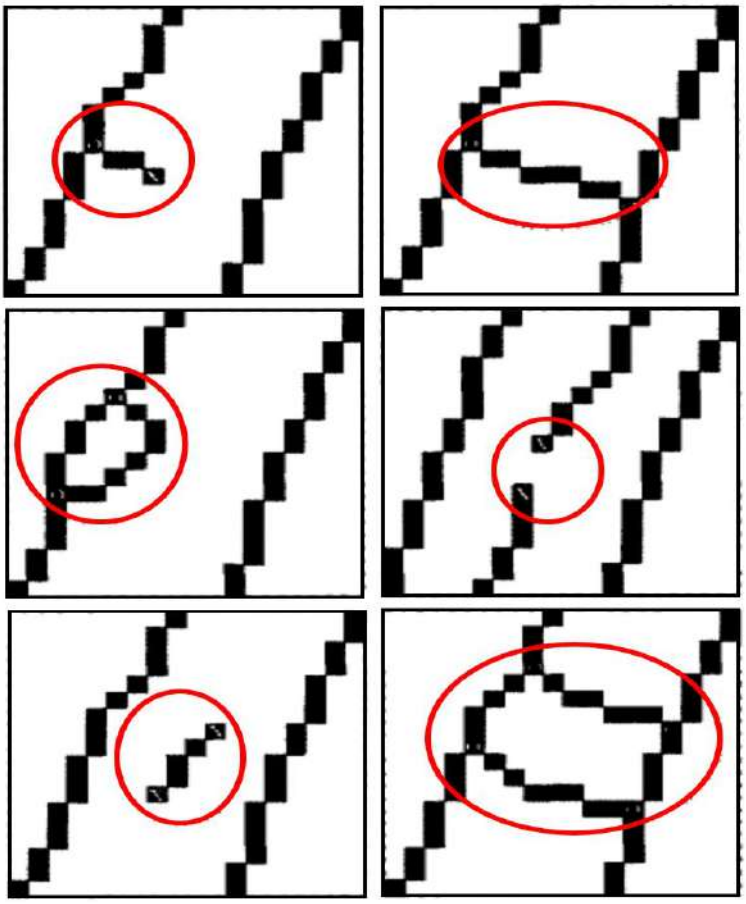

FIGURE 6.1: False minutiae types: From left to right and top to bottom, we have: spike, bridge, hole, break, spur, and ladder [62]

Different researchers have tried to solve the problem of false minutiae before [63–65]. Proposed methods use statistical and structural approaches to remove false minutiae. As a simpler approach adopted by [7, 48], the following techniques are employed to detect and remove false minutiae illustrated in Figure 6.1:

- Since each minutia is expressed as (x, y, θ), if two minutiae are close to each other and have inconsistent θ values then this means they are false minutiae introduced due to discontinuities in ridges and are removed as a post-processing step.
- Minutiae along the boundary of the palmprint are removed.
- Minutiae belonging to a region with a dense population of minutiae are removed. This is because the region probably has too many broken ridges and the minutiae extracted are false.

- Ridge endings too close to ridge bifurcations are removed

Instead of inspecting each minutia individually, some classical minutiae-based methods [7, 47, 48, 50–55] take into account the following properties in small palmprint regions:

- Variance of ridge curvature
- Variance of grayscales, and
- Eligibility of ridges

A high variance of ridge curvature in an $L \times W$ region means rapidly changing ridge orientation depicting poor ridge quality. Similarly, a low variance of grayscales means poor ridge/valley contrast. And eligibility of ridges is usually estimated by calculating the minimum length of ridges. Based on these regional properties, minutiae are discarded or accepted. Results of the above-mentioned techniques are illustrated in Figure 6.2 which shows a portion of the thenar region of a palmprint. It can be seen that some false minutiae have been removed after post-processing. But still, there are a lot of minutiae present in the palmprint that have no contribution in identifying the palmprint and only increase the computational cost of the matching process.

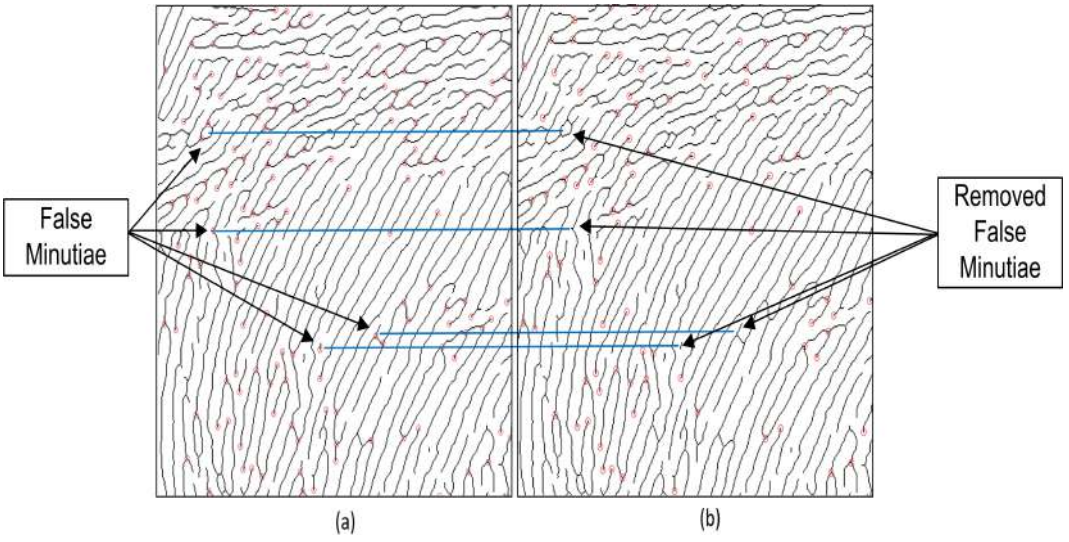

FIGURE 6.2: Portion of thenar region(a) original extracted minutiae, (b) some minutiae removed using post-processing techniques [7, 48]

Generally, minutiae extracted from the hypothenar region are most reliable because of less number of creases, whereas minutiae extracted from the thenar region are least reliable due to the presence of a large number of creases [7]. Figure 6.3 shows minutiae found in the thenar and hypothenar regions of the palm after post-processing illustrated in Figure 6.2. In Figure 6.3, minutiae orientation (θ) is depicted by red arrows. It is can be seen that minutiae extracted from the thenar region exhibit inconsistent orientation and are hence, mostly false.

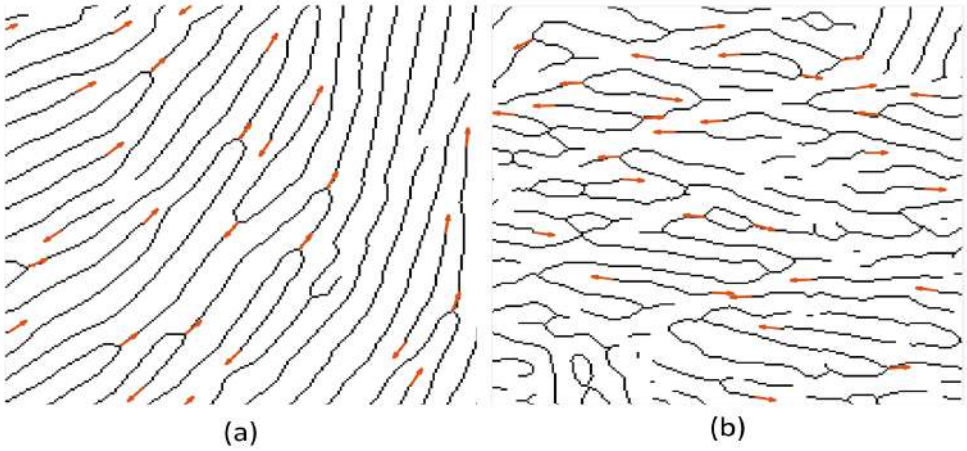

FIGURE 6.3: Mintuiae Orientation: (a) hypothenar region (b) thenar region

Regardless of the efficiency of minutiae removal methods, palmprints still pose a tough computational challenge. This is because palmprints produce 8 times more minutiae than fingerprints [7]. Figure 6.4 illustrates the generic minutiae matching process. As shown in Figure 6.4, each minutia m_i (where $i = 1, 2, ..., N$) in the query image has to be compared with all minutiae $m_j^{'}$ (where $j = 1, 2, ..., M$) of a template image. This means a minimum of $M \times N \times k^2$ matches are required to match two palmprints, where k is the number of neighboring minutiae of each reference minutiae m_i of query palmprint that needs to match with neighbors of each minutia $m_j^{'}$ of a template palmprint (as already explained section 5.3). It is evident that computation requirements for matching two palmprints are directly proportional to the number of minutiae in the palmprints and the value k. As a result, it becomes important to reduce the number of minutiae in the palmprints by only selecting reliable (real) minutiae for the matching stage. A dedicated algorithm for minutiae selection is required that effectively reduces computational requirements by:

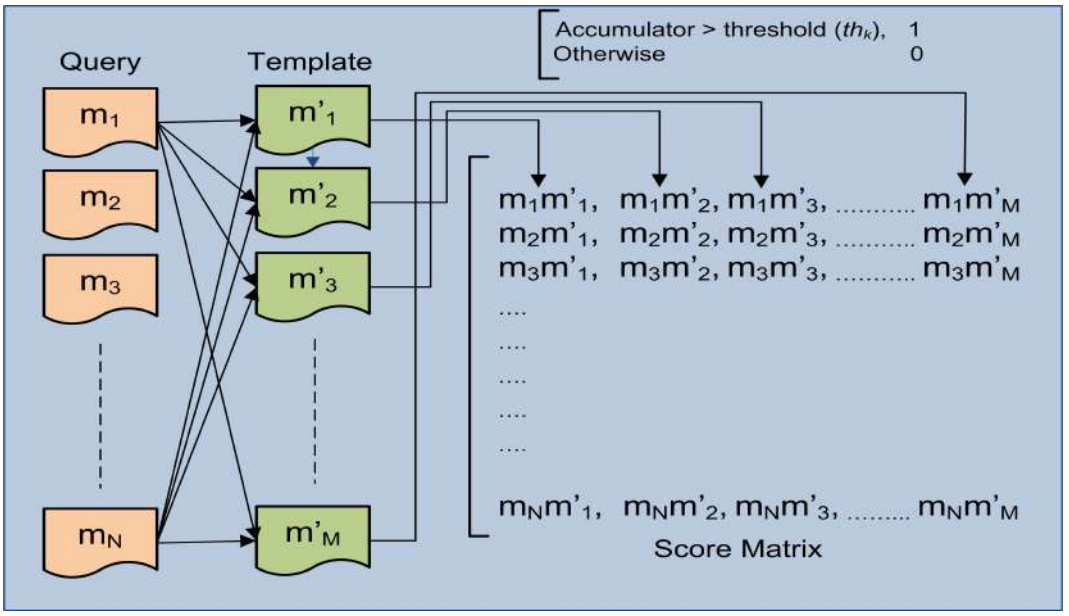

FIGURE 6.4: Generic minutiae matching process

- Identifying and removing false minutiae
- Selecting only a subset out of reliable minutiae to reduce computational overhead

6.2 Proposed Minutiae Selection Algorithm (*MSA*)

The proposed Minutiae Selection Algorithm (*MSA*) is aimed at shortlisting only a subset of minutiae extracted from a palmprint that has the potential of producing good matching accuracy at a reduced computational cost. *MSA* is designed to act as a precursor to minutiae encoding and matching. The proposed algorithm uses a simple histogram-based technique that tries to find the best candidate minutiae for matching using basic minutia characteristics, i.e., spatial coordinates (x, y) and orientation θ. Let's say a palmprint template stored in a database has M number of minutiae associated with it. The proposed method selects a subset M_{sub} out of M minutiae that will be used in the matching stage and exclude the rest from the matching process.

The main idea in the proposed algorithm is that although (x, y, θ) properties of minutiae cannot be directly used to classify minutiae as false or true, differences of orientation θ and (x, y) coordinates between minutiae of a query (input) and

a template palmprint can be used to shortlist strong candidates for the matching process. This is accomplished by storing differences of (x, y, θ) between query and template minutiae in an array and creating histograms of differences **(HoD)**. Peaks in these **HoDs** are identified in an iterative process to shortlist minutiae of choice. A step-wise explanation of *MSA* is given below and further illustrated in Figure 6.5:

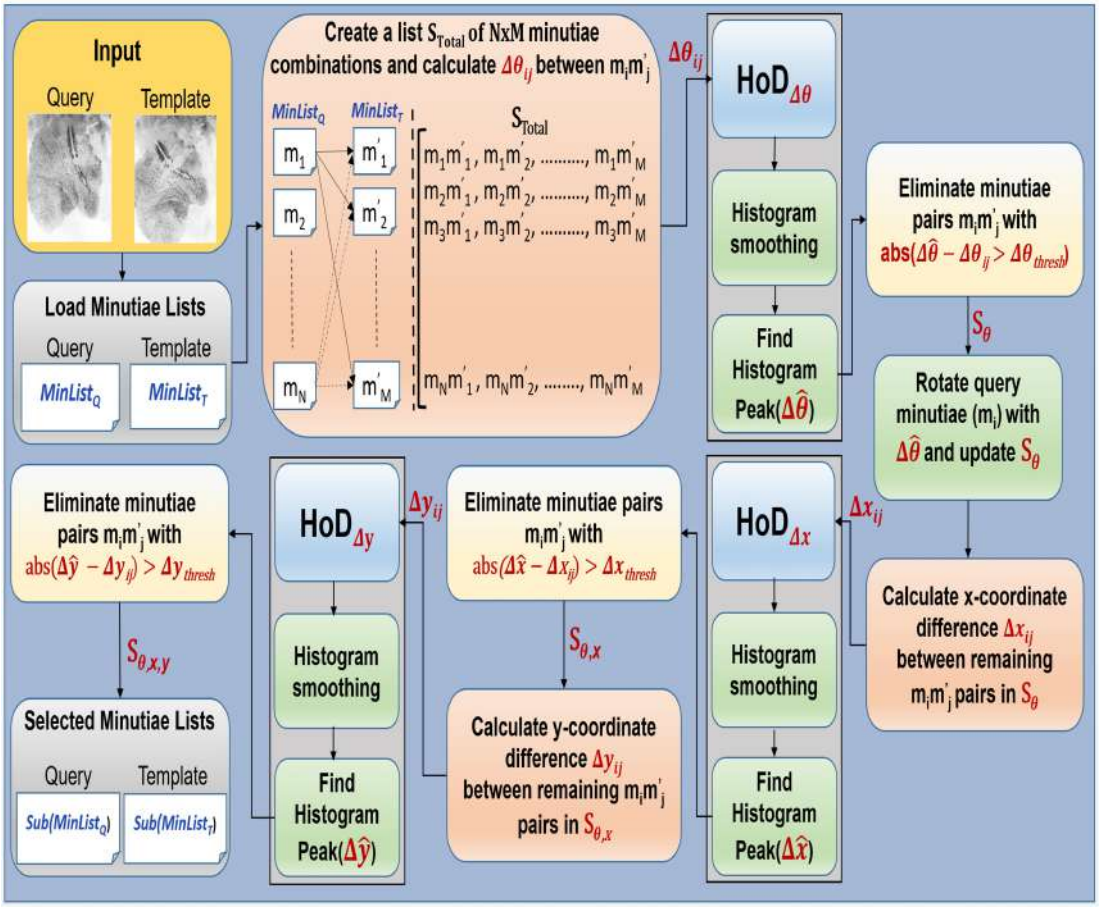

FIGURE 6.5: Minutiae Selection Algorithm (*MSA*): A list S_{Total} containing minutiae pairs $m_i m'_j$ is created. Differences of orientation θ between $m_i m'_j$ are calculated ($\Delta\theta_{ij}$) and a histogram of difference ($\mathbf{HoD}_{\Delta\theta}$) is created. The peak of this histogram ($\Delta\hat{\theta}$) shows the dominant $\Delta\theta$ between minutiae pairs and corresponds to the true orientation difference between the query and template. Minutiae pairs $m_i m'_j$ with $\Delta\theta_{ij}$ greater than $\Delta\theta_{thresh}$ are eliminated from the list yielding a reduced list S_θ. At this stage, query minutiae m_i in S_θ are rotated with the angle $\Delta\hat{\theta}$ and S_θ is updated with new (x, y) values for m_i. In the updated list S_θ, a similar elimination process follows for translation difference Δx_{ij} between minutiae yielding a further reduced list $S_{\theta,x}$. Consequently, a similar elimination process follows for translation difference Δy_{ij} yielding the final reduced list $S_{\theta,x,y}$. Minutiae in the $S_{\theta,x,y}$ are significantly fewer than S_{Total} and give better matching accuracy.

Minutiae Selection Algorithm (*MSA*)

Step-1: Load query minutiae list $MinList_Q = (m_1, m_2, ..., m_N)$ and template minutiae list $MinList_T = (m_1^{'}, m_2^{'}, ..., m_M^{'})$, where $m_i = (x_i, y_i, \theta_i)$; $i = 1, 2, ..., N$, and $m_j^{'} = (x_j, y_j, \theta_j)$; $j = 1, 2, ..., M$. Create a list S_{Total} of $N \times M$ combinations containing minutiae IDs, i, j, and associated (x, y, θ) values, i.e.

$$S_{Total} = \{(i, j, (x_i, y_i, \theta_i), (x_j, y_j, \theta_j))|i \in \{1, 2, ..., N\}, j \in \{1, 2, ..., M\}\}. \tag{6.1}$$

Orientation comparison between query and template minutiae:

Step-2: Find the angular difference between each minutiae pair $m_i m_j^{'}$ in S_{Total} as follows:

$$\Delta\theta_{ij} = \theta_i - \theta_j. \tag{6.2}$$

Step-3: Create a histogram of differences (**HoD**) for all values of $\Delta\theta_{ij}$ (depicted by $\mathbf{HoD}_{\Delta\theta}$ in Figure 6.5) and smooth it by convolving it with a Gaussian smoothing filter. For smoothing, we have used a 21-point Gaussian window with a standard deviation of 4.

$$\mathbf{HoD}_{\Delta\theta}(l) = \sum_i \sum_j \delta(\Delta\theta_{ij} - l), \text{for } l = [-179^o, 180^o], \tag{6.3}$$

$$\overline{\mathbf{HoD}}_{\Delta\theta}(l) = \mathbf{HoD}_{\Delta\theta}(l) * g(z). \tag{6.4}$$

Here $g(z)$ is a gaussian smoothing filter, and δ is defined as

Continued on next page

Minutiae Selection Algorithm (*MSA*)

$$\delta(\Delta\theta_{ij} - l) = \begin{cases} 1, & \text{if } \Delta\theta_{ij} = l \\ 0, & \text{otherwise} \end{cases}$$

Step-4: Find the peak of $\overline{\mathbf{HoD}}_{\Delta\theta}(l)$ as follows:

$$\Delta\hat{\theta} = arg_l max(\overline{\mathbf{HoD}}_{\Delta\theta}(l). \tag{6.5}$$

Since the difference between true minutiae will be constant and most frequent in $\overline{\mathbf{HoD}}_{\Delta\theta}(l)$, the value of l corresponding to the peak in $\overline{\mathbf{HoD}}_{\Delta\theta}(l)$, will represent the orientation difference between the true minutiae of both palmprints and also serves as an overall measure of the orientation difference between palmprints.

Step-5: Define a threshold $\Delta\theta_{thresh}$ on both sides of the peak $\Delta\hat{\theta}$. Create a reduced minutiae list S_θ by eliminating all minutiae combinations S_{Total} with $\Delta\theta_{ij}$ greater than $\Delta\theta_{thresh}$, i.e.,

$$S_\theta = \{(i, j, (x_i, y_i, \theta_i), (x_j, y_j, \theta_j))|(\Delta\hat{\theta} - \Delta\theta_{thresh}) < \Delta\theta_{ij} < (\Delta\hat{\theta} + \Delta\theta_{thresh})\}. \tag{6.6}$$

Step-6: At this point rotate query minutiae in the reduced minutiae list S_θ with the angle $\Delta\hat{\theta}$

$$\tilde{x}_i = x_i cos(\Delta\hat{\theta}) - y_i sin(\Delta\hat{\theta}), \tag{6.7}$$

$$\tilde{y}_i = x_i sin(\Delta\hat{\theta}) + y_i cos(\Delta\hat{\theta}). \tag{6.8}$$

Translation comparison between query and template minutiae:

Step-7: For the reduced minutiae list S_θ, calculate x-component difference between minutiae pairs, i.e., Δx_{ij}:

Continued on next page

Minutiae Selection Algorithm (*MSA*)

$$\Delta x_{ij} = \tilde{x}_i - x_j, \qquad i, j \in S_\theta, \tag{6.9}$$

and create a smoothed histogram $\overline{\mathbf{HoD}}_{\Delta x}$ for Δx_{ij}.

$$\mathbf{HoD}_{\Delta x}(l) = \sum_i \sum_j \delta(\Delta x_{ij} - l) \qquad \text{for } l \in [\Delta x_{min}, \Delta x_{max}], \tag{6.10}$$

$$\overline{\mathbf{HoD}}_{\Delta x}(l) = \mathbf{HoD}_{\Delta x}(l) * g(z). \tag{6.11}$$

Step-8: Now find the dominant x-translation by finding the peak of $\overline{\mathbf{HoD}}_{\Delta x}(l)$,

$$\Delta \hat{x} = arg_l max(\overline{\mathbf{HoD}}_{\Delta x}(l)). \tag{6.12}$$

Step-9: Define a threshold Δx_{thresh} on both sides of the peak $\Delta \hat{x}$. Create a further reduced minutiae list $S_{\theta,x}$ by eliminating all minutiae combinations $m_i m_j^{'}$ with Δx_{ij} greater than Δx_{thresh}, i.e.,

$$S_{\theta,x} = \{(i, j, (\tilde{x}_i, \tilde{y}_i), (x_j, y_j) | i, j, \in S_\theta \cap (\Delta \hat{x} - \Delta x_{thresh}) < \Delta x_{ij} < (\Delta \hat{x} + \Delta x_{thresh})\}. \tag{6.13}$$

Step-10: In the same way, create a further reduced list by comparing y-translation of the remaining minutiae pairs $m_i m_j^{'}$ in $S_{\theta,x}$, i.e,

$$S_{\theta,x,y} = \{(i, j, (\tilde{x}_i, \tilde{y}_i), (x_j, y_j) | i, j, \in S_{\theta,x} \cap (\Delta \hat{y} - \Delta y_{thresh}) < \Delta y_{ij} < (\Delta \hat{y} + \Delta y_{thresh})\}. \tag{6.14}$$

Continued on next page

Minutiae Selection Algorithm (*MSA*)

The final minutiae list $S_{\theta,x,y}$ with their original coordinates and angles are the shortlisted minutiae for matching.

Step-11: By iteratively applying thresholds on $\Delta\theta_{ij}$, Δx_{ij}, and Δy_{ij}, size of the array S_{Total} is considerably reduced to $S_{\theta,x,y}$. This means during matching, instead of carrying out $N \times M$ matches (Figure 6.4), fewer matches need to be performed which reduces the computational cost remarkably and at the same time improves matching accuracy. Both these claims are quantified in the proceeding sections.

Proposed *MSA* is an iterative method of eliminating false or unreliable minutiae simultaneously from query and template palmprint. By applying thresholds on $\mathbf{HoD}_{\Delta\theta}$, $\mathbf{HoD}_{\Delta x}$, and $\mathbf{HoD}_{\Delta y}$ in an ordered fashion, all minutiae combinations $m_i m_j^{'}$ outside $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} are eliminated resulting in a decrease in the size of $m_i m_j^{'}$ combinations. The remaining minutiae combinations $m_i m_j^{'}$ fall within $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh}. This subset of $MinList_Q$ and $MinList_T$ can be then forwarded to the matching stage. Figure 6.6 shows minutiae selected from the thenar region of two palmprints taken from the same subject (genuine match). Minutiae indicated by red circles are the ones shortlisted by the proposed algorithm, whereas green circles indicate minutiae eliminated by the proposed method.

6.2.1 Time and Space Complexity Analysis

Time and space complexity is the most popular way of analyzing an algorithm. The time complexity of an algorithm is defined as the time taken by the algorithm to run as a function of the input size. Similarly, space complexity is the space in

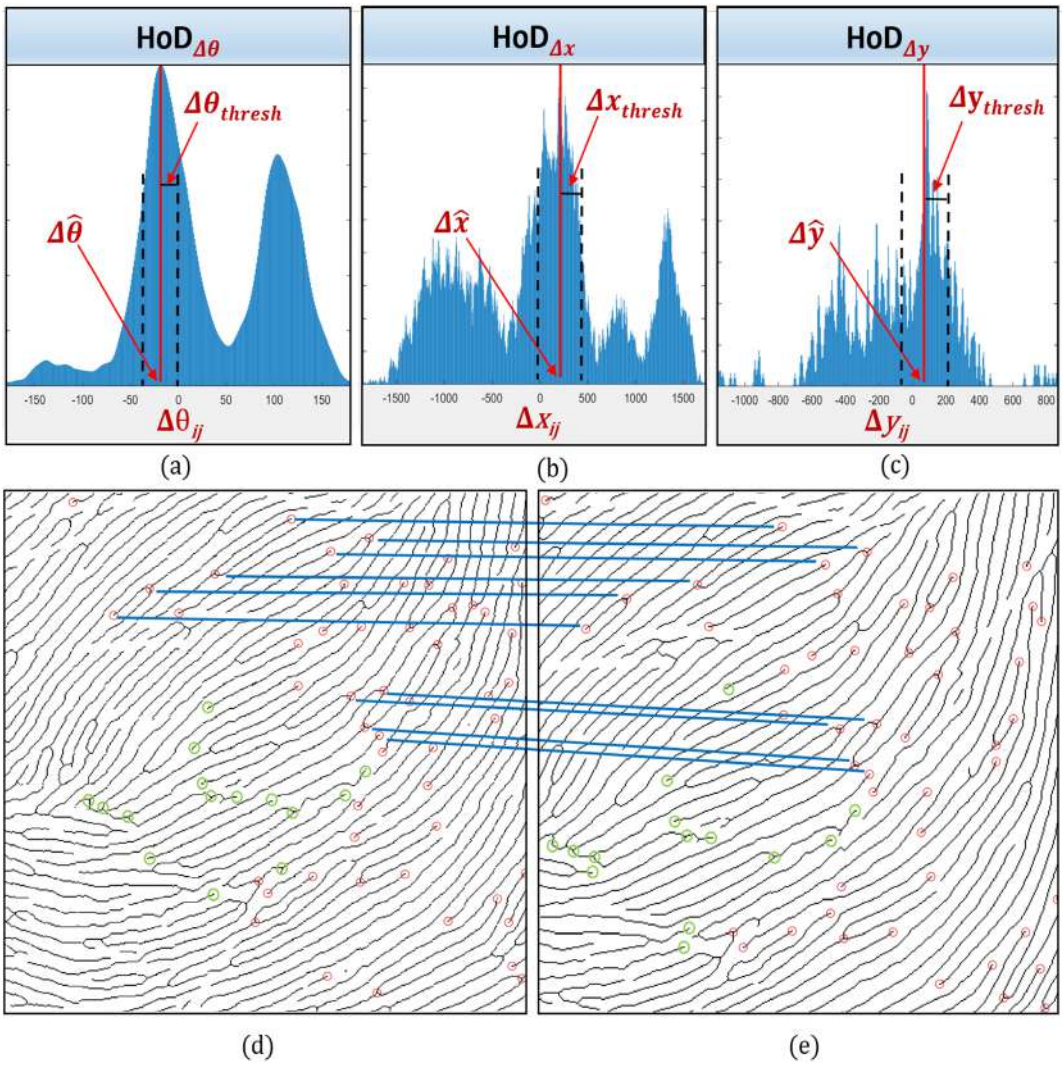

FIGURE 6.6: *MSA* use case: (a),(b), and (c) are $\mathbf{HoD}_{\Delta\theta}$, $\mathbf{HoD}_{\Delta x}$, and $\mathbf{HoD}_{\Delta y}$ respectively. (d) and (e) are patches from two palmprints taken from the same subject with selected minutiae indicated by red circles and eliminated minutiae indicated by green circles

the memory required by the algorithm described as a function of the input size. Although practical time and space calculations depend on a number of different variables such as hardware, operating system, processor cores etc., only the input size is considered to describe the time and space complexity. The execution time of each step is considered to be constant.

In order to represent the time and space complexity of the proposed algorithm, we use the variable n to denote the size of the input to any step of the algorithm and assume a constant running time for each process for simplicity. The process of creating S_{Total} represented by equation 6.1 requires running $N \times M$ times and has a quadratic time complexity of $O(n^2)$. The runtime of the processes represented

by equations 6.2, 6.3, 6.5, and 6.6 grows linearly with the size of the input n. The cumulative time complexity of these 4 processes can be denoted by $4O(n)$. Process of 1d convolution (equation 6.4) using an FFT-based algorithm has a time complexity of $O(nlogn)$. Whereas, the minutia rotation process represented by equations 6.7 and 6.8 in step 6 involves the multiplication of a 1×2 minutia vector (x, y) with a 2×2 rotation matrix. The time complexity of this multiplication for a single minutia is denoted by $O(1 * 2 * 2)$ and this process runs for all values of the input n. Assuming a large value for n, the time complexity of step-6 can be simplified to $O(n)$ only. The processes represented by equations 6.2 through equation 6.6 are repeated 2 more times, i.e., once for Δx and once for Δy (equations 6.9 through 6.14). The overall time complexity of *MSA* can be calculated as:

$$T(n) = O(n^2) + 3O(nlogn) + 13O(n). \tag{6.15}$$

In equation 6.15, the term $O(n^2)$ has a much large growth rate as compared to $O(nlogn)$ and $O(n)$. So the overall time complexity of *MSA* can be simplified to $O(n^2)$.

We follow similar steps to compute the space complexity of *MSA*. Step-1 needs a storage space proportional to $N \times M$ and has a space complexity of $O(n^2)$. Cumulative space complexity for processes represented by equations 6.2, 6.3, and 6.4 can be simplified to $3O(n)$ as it grows linearly with the size of input data. For equation 6.5, space complexity is $O(1)$. Equation 6.6 has no space requirement and for equations 6.7 and 6.8, space complexity can be simplified to $O(n)$. Since, the processes represented by equations 6.2 through equation 6.6 are repeated 2 more times, i.e., once for Δx and once for Δy (equations 6.9 through 6.14). the overall space complexity of *MSA* can be calculated as in equation 6.16. Like time complexity, space complexity of *MSA* can also be simplified to $O(n^2)$:

$$S(n) = O(n^2) + 9O(n) + 3O(1). \tag{6.16}$$

6.3 Conclusion

This chapter provides background, concept, and implementation details of the proposed *MSA* along with time and space complexity analysis. *MSA* has both online and offline applications. Both types of applications and their effect on computation and identification accuracy are discussed in the proceeding chapter 7.

Chapter 7

Results and Applications of the Proposed Minutiae Selection Algorithm (*MSA*)

Experiments have been conducted on THUPALMLAB high resolution dataset [44]. To the best of our knowledge, this challenging dataset has been used in all state-of-the-art studies on high resolution palmprints. The dataset consists of 1280 palmprint images taken from both the left and right hands of 80 different subjects, with eight impressions per hand. The images have a resolution of 2040 $\times$ 2040 pixels at 500 ppi with 256 grey levels. n-nearest neighbor matching [103] implemented by Ghafoor et al. [48] has been applied on the minutiae shortlisted by *MSA*. In the n-nearest method, each minutia is encoded based on its n nearest neighboring minutiae. The value of n is chosen to be 10. An overview of the matching scheme used [48] is as follows:

For a reference minutia m_i, encoding $en(m_i)$ based on n nearest neighbors m_k is done using r_{ik}: distance between m_i and m_k, $\Delta\theta_{ik}$: angle difference between m_i and m_k, and ϕ_{ik}: angles of the edges joining m_i and m_k. This implies that a reference minutia m_i is encoded as follows:

$$en(m_i) = \{r_{ik}, \Delta\theta_{ik}, \phi_{ik}\}. \tag{7.1}$$

Here r_{ik}, $\Delta\theta_{ik}$, and ϕ_{ik} are calculated as:

$$r_{ik} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}, \tag{7.2}$$

$$\Delta\theta_{ik} = (\theta_i - \theta_k), \ and \tag{7.3}$$

$$\phi_{ik} = tan^{-1}\left(\frac{y_i - y_k}{x_i - x_k}\right), \tag{7.4}$$

Here $n = 1, 2, ..., 10$. Encoded minutiae are then stored as a template of the palmprint in the database.

$$En(N) = \{en(m_1), en(m_2), ..., en(m_N)\}. \tag{7.5}$$

During matching, each minutia of a query (input) palmprint is matched with all minutiae of template palmprints using an exhaustive point-pattern search algorithm as shown in Figure 6.4. Since each minutia is encoded using n-nearest neighbors, matching one minutia pair $m_i m_j^{'}$ requires n^2 feature comparisons. For a minutia in query palmprint to match with a minutia in template palmprint, at least th_k amount of neighbors out of n need to match. The number of matching neighbors (A_{ij}) between a minutia pair $m_i m_j^{'}$ are calculated and stored in a score matrix S as

$$S(i,j) = \begin{cases} 1, & \text{if } A_{ij} \geq th_k \\ 0, & \text{otherwise} \end{cases}$$

In all the experiments, the value of th_k is fixed at 5. Similarly, the score matrix S values for each minutia pair $m_i m_j^{'}$ are calculated. Finally, the consolidated percentage similarity score between two palmprints is evaluated by computing the ratio of matched minutiae pairs in score matrix S and the total number of minutiae pairs as

$$Similairty = \frac{minutia\_pairs\_matched}{min(M,N)} \times 100. \tag{7.6}$$

Performance of *MSA* is judged in two domains, i.e., a significant reduction in minutiae matches ($N \times M$) and improvement in matching accuracy. Specifically, it is shown that changing the values of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} has a significant effect on the number of minutiae eliminated and resulting matching accuracy. Through contrast analyses, it is shown that tight values of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} give a greater reduction in the number of minutiae matches but deteriorate matching performance. On the other hand, lose values of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} give smaller reduction in minutiae matches.

In order to calculate matching accuracy, the metric of Equal Error Rate (**EER**) is used. EER is calculated by computing the False Matching Rate (**FMR**) and the False Non-Matching Rate (**FNMR**). EER represents a threshold where both FMR and FNMR are equal. FNMR (also called genuine matching) is determined by comparing each palmprint sample of a hand of a subject with other samples of the same subject's hand. The total number of genuine matches is calculated as $((8\times7)/2)\times80\times2 = 4480$. FMR (also called impostor matching) is determined by comparing the first palmprint sample of each hand of a subject with the first palmprint sample of the same hand of the remaining subjects. The total number of imposter matches is calculated as $(80\times79)/2)\times2=6320$. Effects of employing *MSA* prior to matching are also illustrated using Detection Error Tradeoff (**DET**) Since *MSA* is designed as a precursor to minutiae matching, it finds its applications in both offline (enrollment) and online stages. Results in both types of applications show that *MSA* allows the matching algorithm to achieve higher accuracy at a reduced computational cost. Details about the offline and online application of *MSA* are given in the proceeding sections with results and discussion.

7.1 Offline Application

Offline application of the proposed minutiae selection algorithm is carried out at the enrollment stage. Enrollment is the stage when persons are registered into a biometric system. During enrollment, each person inputs multiple samples from

the same palm. Each palmprint is then enhanced, and features are extracted, encoded, and saved in the database as templates. *MSA* is employed before encoding the extracted minutiae. *MSA* is able to reduce the total number of minutiae that need to be encoded and stored in database templates during the offline or enrollment stage. This reduces computation requirements during the online matching stage because fewer minutiae need to be matched.

Let's assume that the database contains a total of **T** subjects and 4 samples per subject (for THUPALMLAB dataset [44], T=80 and there are 8 samples per palm for each subject). After initial minutiae extraction, there is a minutiae list M_{pq} corresponding to each palmprint, where p indicates the subject, i.e., $p = (1, 2, ..., T)$ and q indicates sample number of each subject p, i.e., $q = (1, 2, 3, 4)$. Minutiae lists M_{pq} extracted from multiple samples of the same subject are compared using the *MSA* to reduce of the number of minutiae that need to be encoded and matched at later stages. Application steps are illustrated in Figure 7.1 and further explained below:

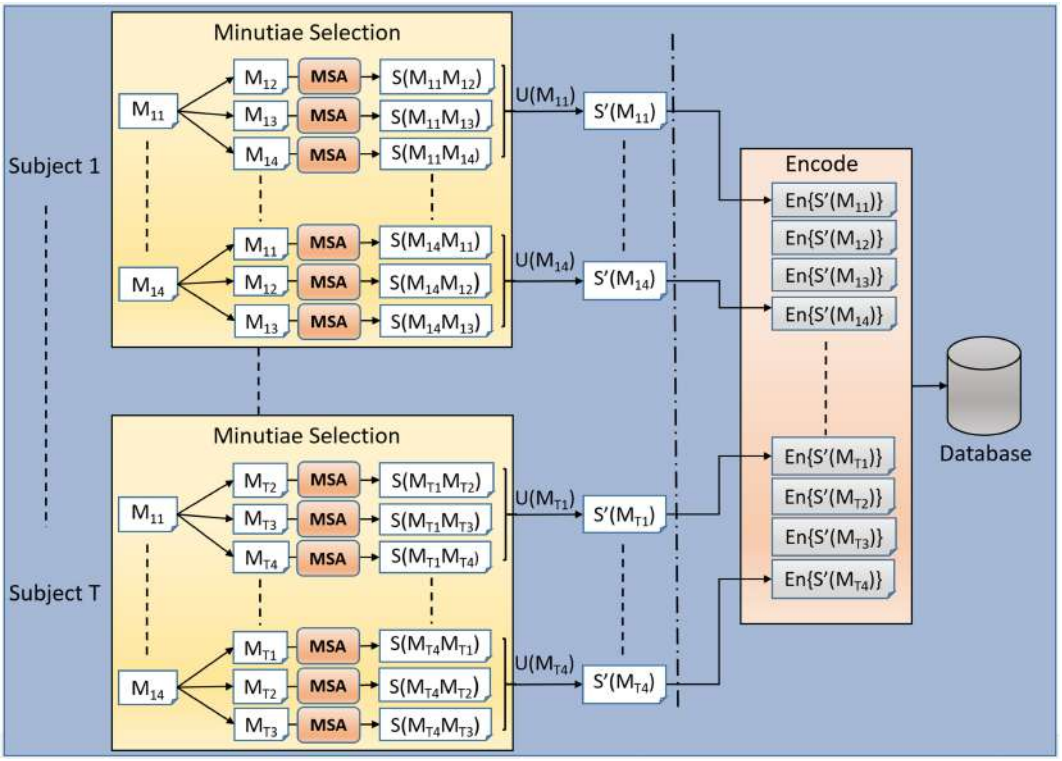

FIGURE 7.1: Offline Application of *MSA*: For each subject, p, where $p = (1, 2, ..., T)$, each sample q is compared with other samples of the same subject p using *MSA* and selected minutiae are shortlisted . For each sample of subject p, minutiae in the union set $U(M_{pq})$ give final selected minutiae $S^{'}(M_{pq})$ which are encoded and stored in the database.

Offline Application of *MSA*

Step-1: For a particular subject p, load minutiae list of the first sample p, i.e., M_{p1}.

Step-2: Match M_{p1} with minutiae lists of remaining samples of the same subject p, i.e., M_{p2}, M_{p3}, and M_{p4} using *MSA* (section **??**). This will give a reduced list of selected minutiae depicted by $S(M_{p1}, M_{p2})$, $S(M_{p1}, M_{p3})$, and $S(M_{p1}, M_{p4})$ (see Figure 7.1).

Step-3: Extract selected minutiae belonging to M_{p1} from $S(M_{p1}, M_{p2})$, $S(M_{p1}, M_{p3})$, and $S(M_{p1}, M_{p4})$ and perform union operation $U(M_{p1})$ to remove repeating minutiae. This gives us the reduced minutiae list $S^{'}(M_{p1})$ (see Figure 7.1) which is the final list of shortlisted minutiae for M_{p1}.

Step-4: Repeat Step-2 and Step-3 for all samples of all subjects $p = (1, 2, ..., T)$. This will give us a set $S^{'}(M_{pq})$ corresponding to all samples of all subjects in the database.

Step-5: Encode $S^{'}(M_{pq})$ using n-nearest neighbor method [103] and store encoded templates $En(S^{'}(M_{pq}))$ in the database.

Step-6: During online matching, carry out minutiae-based matching (Figure 6.4) using $En(S^{'}(M_{pq}))$ templates.

The fact that $S^{'}(M_{pq})$ contains a considerably smaller amount of minutiae as compared to M_{pq}, minutiae-based matching in the online stage is computationally more efficient. Also due to the elimination of most false minutiae by employing *MSA*, matching accuracy also improves. Results of selecting potentially better

candidate minutiae through *MSA* at the offline stage are presented in Table 7.2. Best results are highlighted in gray.

$\Delta\theta_{thresh}$	Δx_{thresh}	Δy_{thresh}	**Reduction in Minutiae Matches**	**EER %**
5	10	10	46%	1.3
10	20	20	25%	0.22
15	30	30	17%	0.01
20	40	40	9%	0.02
25	50	50	5%	0.04

TABLE 7.2: Results of Offline *MSA* Application using different configurations of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh}

Table 7.2 lists results of employing *MSA* at the offline stage in terms of reduction in minutiae matches and matching accuracy. It can be seen that by varying values of *MSA* parameters, i.e., $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} superior performance can be achieved. Tighter values of these parameters yield a larger reduction in minutiae matches but deteriorate matching performance, while relaxed values show higher matching accuracy but provide a smaller reduction in minutiae matches. Since n-nearest neighbor matching method [103] implemented by Ghafoor et. al. [48] has been used, the efficacy of offline application of *MSA* can be easily established by comparing the performance of the matching algorithm with and without *MSA*.

Method	***MSA* parameters**	**EER %**	**Details**
Ghafoor et al.[48]	-	0.18	Without *MSA*
Ghafoor et al.[48]	$\Delta\theta_{thresh}$=10, Δx_{thresh}=20, Δy_{thresh}=20	0.22	With MSA, 25% fewer minutiae
Ghafoor et al.[48]	$\Delta\theta_{thresh}$=15, Δx_{thresh}=30, Δy_{thresh}=30	0.01	With MSA, 17% fewer minutiae

TABLE 7.3: Matching accuracy with and without *MSA* (Offline)

Table 7.3 shows that the same matching algorithm is able to perform better by choosing better minutiae using *MSA*. *MSA* offers the opportunity of striking a balance between accuracy and computational efficiency. For example with

$\Delta\theta_{thresh}$=10, Δx_{thresh}=20, and Δy_{thresh}=20, *MSA* is able to provide almost same EER% with 25% fewer minutiae matches. On the other hand, with $\Delta\theta_{thresh}$=15, Δx_{thresh}=30, and Δy_{thresh}=30, *MSA* achieves a remarkable accuracy with EER value of 0.01 with 17% fewer minutiae matches. In order to further corroborate the proposed *MSA*, this EER value has been compared with EER values achieved in state-of-the-art studies in Table 7.4. Table 7.4 shows that the use of *MSA* enables the matching algorithm to fare better as compared to most state-of-the-art.

Method	**EER%**
Jain et al. [7]	5.04
Dai and Zhou [56]	2.99
Capelli et al. [104]	0.01
Tariq et. al. [47]	0.38
Hussain et. al. [58]	0.04
Khodadoust et. al. [14]	2.01
Ghafoor et. al. [48]	0.18
Ghafoor et. al. [48] with *MSA*	0.01

TABLE 7.4: Matching accuracy comparison with state-of-the-art using *MSA* (Offline)

The next section argues the efficacy of employing *MSA* at the online stage.

7.2 Online Application

This application of *MSA* is carried out during the online matching stage. The input query palmprint is pre-processed and minutiae are extracted from it, i.e. M_Q. The query minutiae list M_Q is then compared using *MSA* with the template minutiae list M_T. This way only a subset of minutiae are selected from both query and template palmprints and sent to the matching stage. Details are illustrated in Figure 7.2 and further explained below:

A stepwise implementation of the online application of *MSA* is explained below:

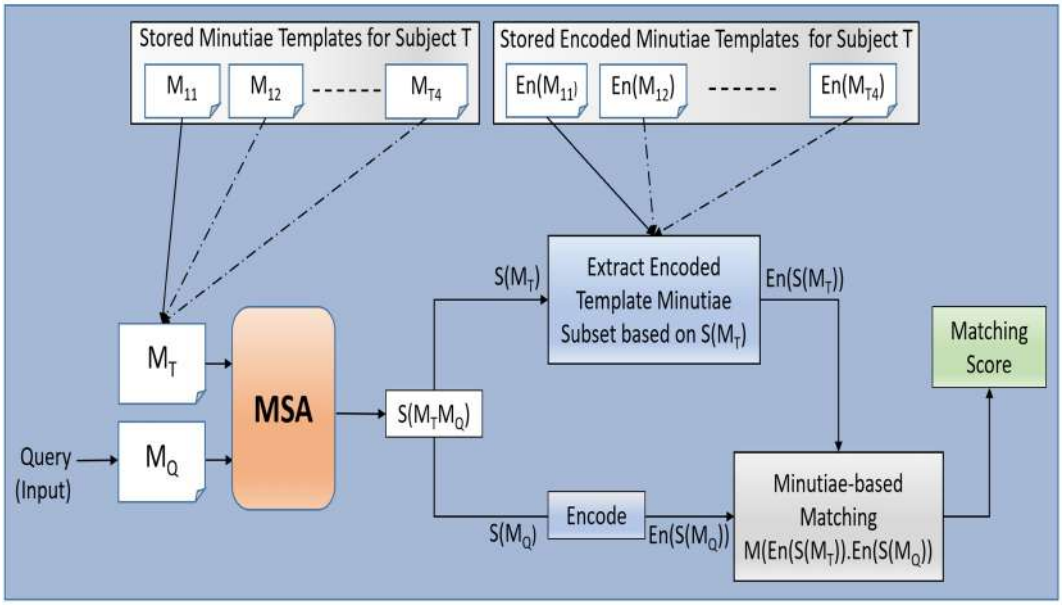

FIGURE 7.2: Online Application of *MSA*: Query minutiae list M_Q alongwith template minutiae list M_T is passed to *MSA* and a subset of minutiae are selected in the list $S(M_Q M_T)$. Subsequently, the corresponding subset of encoded minutiae templates is used to match both palmprints.

Online Application of *MSA*

Step-1: Pass M_Q and M_T as inputs to the *MSA*. This outputs a reduced list depicted as $S(M_Q M_T)$ in Figure 7.2.

Step-2: Extract $S(M_Q)$ and $S(M_T)$ from the list. Forward $S(M_Q)$ to the online matching stage after encoding, i.e., create $En(S(M_Q))$.

Step-3: Based on IDs of the minutiae shortlisted in $S(M_T)$, extract $En(S(M_T))$ from $En(M_T)$ (stored in the database) and forward to the online matching stage.

Step-4: Based on encoded minutiae subsets $En(S(M_T))$ and $En(S(M_Q))$ calculate similarity score between query and template palmprint using equation 7.6.

Since $S(M_T)$ and $S(M_Q)$ contain a considerably smaller amount of minutiae as compared to M_T and M_Q, minutiae-based matching in the online stage is computationally more efficient in terms of computational cost. Results of online application of *MSA* using different values of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} are listed in Table 7.6. It can be observed that the values of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} are a lot more relaxed in the online application as compared to offline application. This is because, in the online application, the query palmprint is matched with a template palmprint in a one-to-one fashion, while in the offline application, each palmprint is compared with all samples of the same subject and union of all minutiae subsets is taken as the final minutiae subset.

$\Delta\theta_{thresh}$	Δx_{thresh}	Δy_{thresh}	**Reduction in Minutiae Matches**	**EER %**
15	30	30	64%	1.26
20	40	40	51%	0.54
25	60	60	38%	0.36
30	80	80	27%	0.12
30	100	100	23%	0.15

TABLE 7.6: Results of Online *MSA* Application using different configurations of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh}

The efficacy of online application of *MSA* can be easily established by comparing the performance of the matching algorithm with and without *MSA* (7.7).

Method	***MSA* parameters**	**EER %**	**Details**
Ghafoor et al.[48]	-	0.18	Without *MSA*
Ghafoor et al.[48]	$\Delta\theta_{thresh}$=30, Δx_{thresh}=80, Δy_{thresh}=80	0.12	With *MSA*, 36% fewer minutiae
Ghafoor et al.[48]	$\Delta\theta_{thresh}$=30, Δx_{thresh}=100, Δy_{thresh}=100	0.15	With *MSA*, 32.5% fewer minutiae

TABLE 7.7: Matching accuracy with and without *MSA* (Online)

By comparing Table 7.3 and Table 7.7, it can be seen that the online application of *MSA* is not able to achieve matching accuracy comparable to the offline application but it still performs favorably as compared to the state-of-the-art methods listed in Table 7.8 with fewer minutiae matches.

Method	**EER%**
Jain et al. [7]	5.04
Dai and Zhou [56]	2.99
Capelli et al. [104]	0.01
Tariq et. al. [47]	0.38
Hussain et. al. [58]	0.04
Khodadoust et. al. [14]	2.01
Ghafoor et. al. [48]	0.18
Ghafoor et. al. [48] with *MSA*	0.12

TABLE 7.8: Matching accuracy comparison with state-of-the-art using *MSA* (Online)

7.3 Discussion

Results achieved using different configurations of *MSA* thresholds in Table 7.2 and Table 7.6 reveal that $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} can be conveniently used to trade-off between accuracy and computational efficiency (number of minutiae matches). Strict thresholds are good at eliminating false minutiae but also have the potential of eliminating some of the true minutiae resulting in an increased false non-match rate (**FNMR**). Hence, although strict thresholds provide good computational efficiency they deteriorate matching performance. On the other hand, relaxed thresholds keep most of the true minutiae intact but end up including false minutiae as well. This not only reduces computational efficiency but the inclusion of false minutiae also adversely affects matching performance by increasing the false match rate (**FMR**). This phenomenon is illustrated below in Figure 7.3.

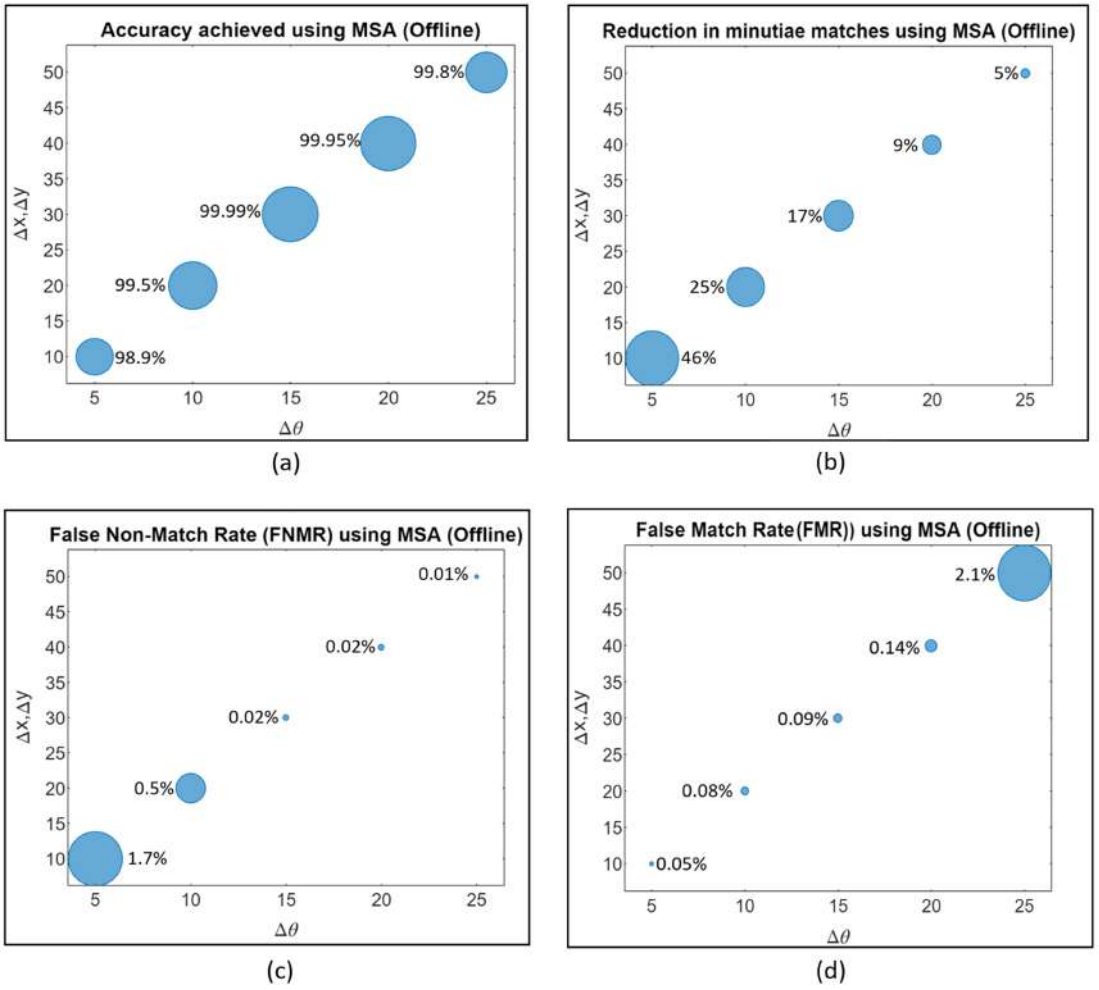

FIGURE 7.3: Effect of *MSA* thresholds (Offline) on (a) matching accuracy, (b) Reduction in minutiae matches (Genuine plus Impostor matches), (c) FNMR, (d) FMR.

The effect of strict vs relaxed configurations of *MSA* thresholds in the offline application has been illustrated by plotting four quantities, namely, accuracy, computational overhead, FMR, and FNMR at various values of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh}. Figures 7.3(a) and 7.3(b) show that keeping *MSA* thresholds tight achieves a greater reduction in minutiae matches but does not provide the best matching accuracy. Similarly, matching accuracy decreases when *MSA* thresholds are relaxed too much. This phenomenon is explained by the increase of FNMR at tight thresholds in Figure 7.3(c) and the increase of FMR at relaxed thresholds in Figure 7.3(d). Both FNMR and FMR contribute towards the overall matching accuracy achieved by the matching algorithm.

Similar behavior can be seen in the online application of *MSA* (Figure 7.4) where a trade-off can be achieved between accuracy and matching speed. As explained

in section 7.2, *MSA* thresholds are kept relaxed as compared to the offline stage due to different matching scenarios. Although *MSA* employment at the online stage achieves slightly lower matching accuracy, it provides a greater reduction in minutiae matches required.

The detection error trade-off (DET) graph is an alternative metric to EER. DET graph plots False Non-Match Rate (FNMR) against False Match Rate (FMR) on a logarithmic scale. DET graphs for offline and online applications of *MSA* using the same encoding and matching algorithm are presented in Figure 7.5. By comparing FMR and FNMR plots of the matching algorithm with and without employing *MSA*, it can be seen that *MSA* significantly improves the performance of the matching algorithm and at the same time reduces the number of minutiae matches required considerably.

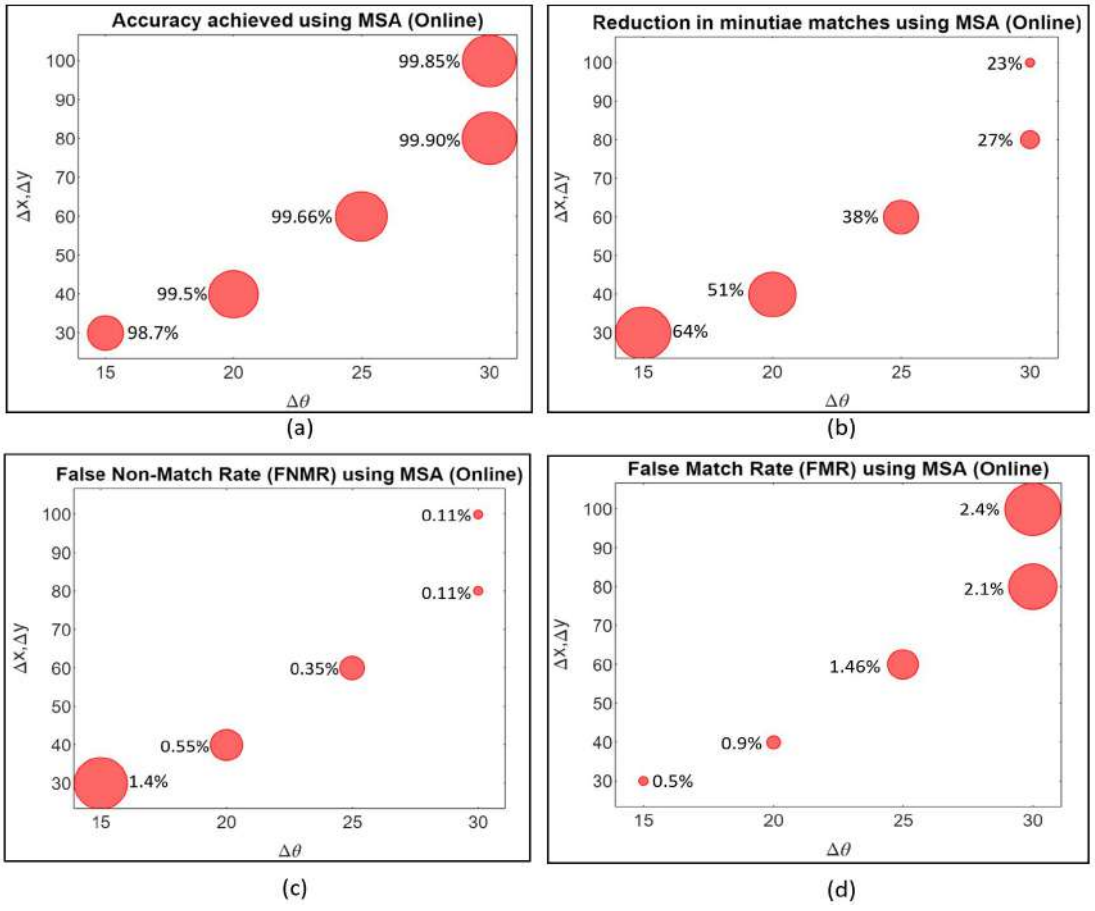

FIGURE 7.4: Effect of *MSA* thresholds (Online) on (a) matching accuracy, (b) Reduction in minutiae matches (Genuine plus Impostor matches), (c) FNMR, (d) FMR.

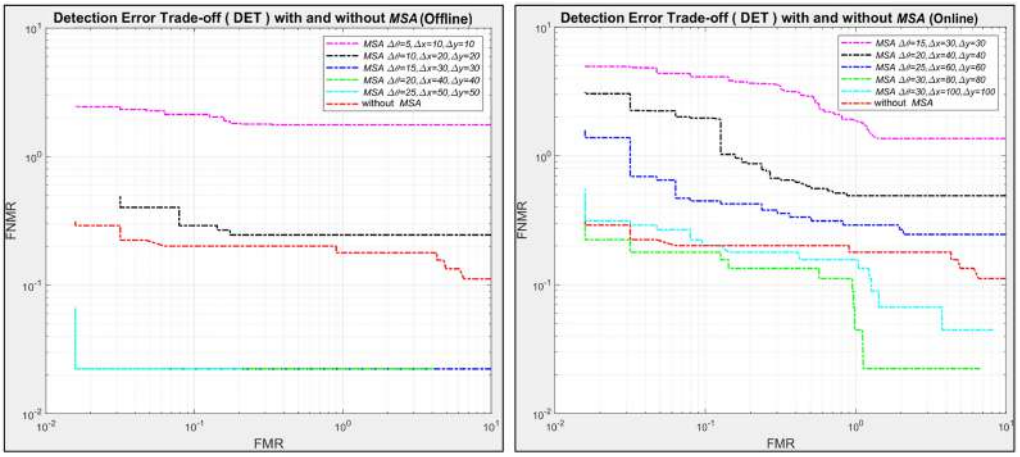

FIGURE 7.5: DET graphs for offline and online applications of *MSA*

In order to arrive at a more precise conclusion regarding the significance of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh}, we use contrast analysis to get additional insight into differences produced by modifying these thresholds on the number of minutiae eliminated. Here combinations of $\Delta\theta_{thresh}$, Δx_{thresh}, and Δy_{thresh} values are taken as the independent variable and their effect on the reduction of minutiae is observed. Through contrast analysis, we are able to show that by changing the values of thresholds, a statistically significant difference between means of eliminated minutiae is observed in both genuine and impostor matches irrespective of the online or offline choice of application. In Figure 7.6, the number of minutiae eliminated in genuine matches using *MSA* are distributed among five groups, which are the five different combinations of *MSA* thresholds. The trend observed in Figure 7.6 is consistent with the hypothesis that the number of minutiae eliminated is reduced as we move from tight to loose *MSA* thresholds. Also, a p-value of ¡ 0.001 presents strong evidence that *MSA* thresholds have a statistically significant effect on the number of minutiae eliminated. Similar trends are observed in impostor matches in Figure 7.7.

It is highlighted that all the results presented in section 7.1, 7.2, and 7.3 are acquired on THUPALMLAB [44] database which is a very challenging high resolution palmprint database. *MSA* thresholds presented in this paper were chosen after sufficient experimentation on the mentioned dataset. *MSA* thresholds can

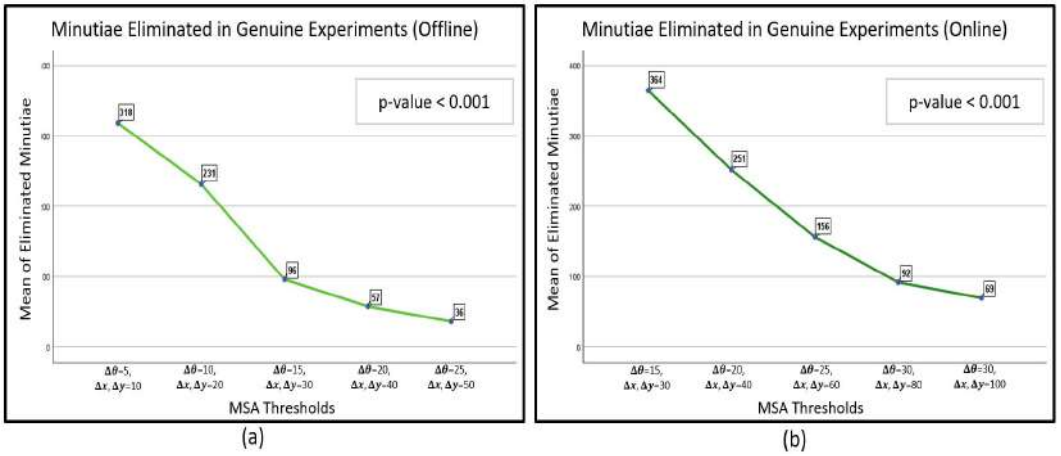

FIGURE 7.6: Contrast Analysis (Genuine Matches): Effect of *MSA* thresholds on means of the number of minutiae eliminated during Genuine matches in Offline and Online Application

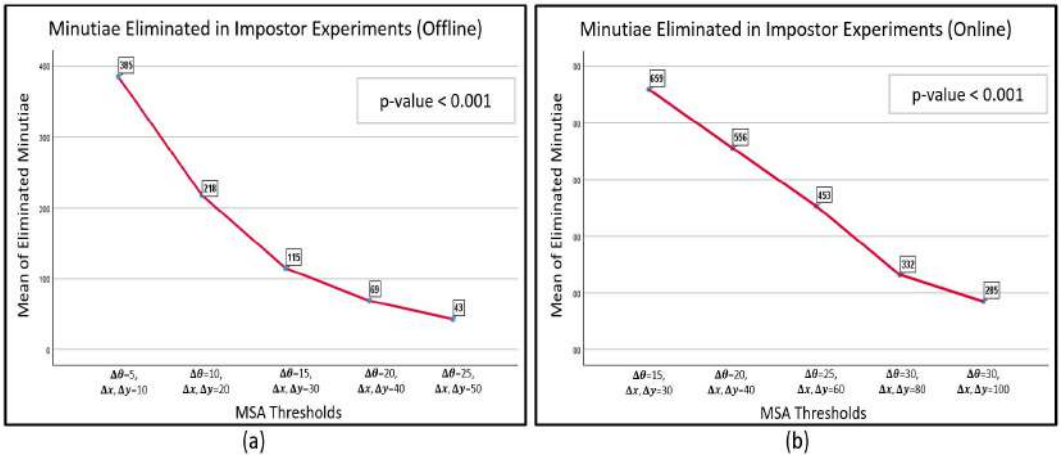

FIGURE 7.7: Contrast Analysis (Impostor Matches): Effect of *MSA* thresholds on means of the number of minutiae eliminated during Impostor matches in Offline and Online Application

be easily tailored according to the dataset, application, minutiae encoding, and matching algorithms as *MSA* is totally independent of these choices.

7.4 Conclusion

The last two chapters propose a simple histogram-based Minutiae Selection Algorithm (*MSA*) that acts as a precursor to minutiae matching. While comparing two palmprints, the proposed method utilizes only the basic properties of minutiae, i.e., (x, y, θ) to shortlist the best candidates in both palmprints for matching. Employment of *MSA* before matching not only reduces the number of minutiae matches

but also improves matching accuracy. Previous research works have employed elaborate pre-processing techniques such as image registration using Generalized Hough Transform (GHT) and CNNs or blockwise matching algorithms to speed up the palmprint matching process. *MSA* does not require image registration and is independent of minutiae encoding and matching methods. This means it can be used in conjunction with any minutiae encoding or matching methods. Furthermore, *MSA* can be applied at both offline and online stages. Results in both applications show that by employing *MSA*, the matching algorithm achieves superior accuracy with reduced computational overhead. In the future, we intend to use *MSA* by aiding it with ridge quality information from the palmprint to further improve its computational and matching performance.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This thesis highlights the recent increase in the worldwide utility of biometric systems. Market indicators and the rise of the research community's interest in biometric systems have been shown to point toward this rising trend. The shifting of more and more consumer services to online systems and ever-rising security concerns have added to the importance of dependable biometric means for human identification. Human face, iris, gait, signatures, fingerprints, and palmprints have been popular choices for large-scale identification systems in the recent past. However, this study points out htat palmprints along with fingerprint-based systems were the leaders in the biometric industry in the last decade.

A literature survey of modern palmprint applications done during the course of this study shows that palmprints are a unique biometric trait as they provide a variety of identification features at low and high image resolutions. While low resolution applications of palmprints give more convenience to users, high resolution palmprints provide higher identification accuracy and security against the stealth of digital identities. Available literature on palmprints also reveals that according to a survey conducted by USA's Federal Bureau of Investigation (FBI), 30% of evidence found in crime scenes is in the form of high resolution palmprints which is conclusive proof that palmprints based systems are the future of the biometric

industry.

This thesis further points out that most methods used for pre-processing, post-processing, and matching high resolution palmprints have been borrowed from the domain of fingerprints. However, despite apparent similarities between fingerprints and high resolution palmprints, fingerprint-based approaches cannot be directly applied to high resolution palmprints. The flexibility of skin, major and minor creases, and poor ridge structure of palmprints acquired from real-world scenarios present challenges during pre-processing steps that are not encountered in fingerprints. As a result, the efficiency of pre-processing steps is greatly deteriorated resulting in poor matching accuracy and increased computational overhead. Recent work on high resolution palmprints has focused more on improving the efficiency of matching algorithms. Surprisingly, limited novelty has been introduced in the pre-processing of palmprints, which include: palmprint region of interest (ROI) segmentation, palmprint enhancement, and feature selection.

After reviewing recent work on palmprint identification systems, this thesis points out that the major problems for any matching algorithm are the large number of features (minutiae) extracted from the palmprint and the fact that most of these features are false due to error-prone enhancement of palmprints. It has been argued that the adverse effects of these factors can be greatly mitigated by employing pre-processing techniques that are specifically designed for palmprints. In order to supplement this claim with quantitative proofs, this thesis presents three separate studies dealing with pre-processing of palmprints.

Firstly, this thesis highlights the performance issues of grayscale variance-based ROI segmentation methods used by previous studies. When employed on palmprints, they do not perform as efficiently due to complex background structures resulting in an image containing unwanted segments of the background. While dealing with real-world palmprints, it is hard to distinguish between foreground and background by calculating the textural differences. After studying the results of previous methods, this thesis proposes a frequency domain ROI Segmentation method specifically designed for palmprints to deal with background textures and additive noise. Instead of estimating the grayscale variance, the proposed method is able to extract the palmar region in the image by isolating only palm ridge

frequencies and removing all others. A comparison drawn with the previous approaches verifies the efficacy of the proposed method. By significantly reducing the area of interest, subsequent processing is restricted to valid regions only. Considering the large size of high resolution palmprints, this significantly reduces the computational overhead of the subsequent processes. Secondly, the possibility of background textures being falsely classified as palm ridges is also eliminated. Since the method is implemented in the frequency domain, it is independent of image contrast and illumination which are generally poor in naturally acquired high resolution palmprints.

Secondly, recent palmprint enhancement techniques have been studied and their shortcomings have also been identified. It has been argued that techniques borrowed from the fingerprint domain donot work as efficiently for palmprints due to the challenging nature of palm ridge structure. Apart from lacking the requisite robustness to deal with palmprint ridge structure, they are computationally exhaustive. In order to tackle both these issues in palmprint enhancement, a robust high resolution palmprint enhancement method is proposed which bypasses traditional enhancement methods completely using deep learning. The proposed enhancement method converts a palm patch directly from the spatial domain to its corresponding enhanced version. Furthermore, the proposed method is able to work on sufficiently large patches of palmprint (96×96) which reduces overall processing time. Palmprint is split into 96×96 patches which are classified by Cnet on the basis of dominant ridge orientation. After prediction by Cnet, patches are rotated (if needed) to align with the kernels of Rnet. Rnet directly converts a patch to its corresponding enhanced version without the need for any ridge orientation or frequency estimate. Rnet produces corresponding enhanced versions of patches which are subsequently joined to produce a complete enhanced palmprint. It has sufficient depth and kernels to accommodate difficult ridge patterns encountered in palmprints due to abruptly changing ridge orientation and various types of noise. The proposed method is totally independent of ridge frequency. By looking at the results of the proposed enhancement method, it can be concluded that it performs favorably in comparison to the state-of-the-art by being able to

recover ridge continuity even in high crease areas. Enhancement results are further corroborated by matching results. Matching results achieved for palmprints enhanced using the proposed method are compared with state-of-the-art and are a testament to the reliability of extracted features.

Lastly, this study also points out that limited effort has been put in to reduce the number of minutiae required to match palmprints. To deal with the issue of a large number of minutiae extracted from palmprints (most of which are false), a minutiae selection algorithm is proposed that shortlists the best candidate minutiae for matching using an intuitive histogram-based algorithm. While previous studies employ additional parameters like ridge quality or local ridge orientation properties to judge the quality and reliability of minutiae, the proposed algorithm utilizes only the basic properties of minutia to make the decision, i.e., (x, y, θ). The main idea in the proposed algorithm is that although (x, y, θ) properties of minutiae cannot be directly used to uniquely identify palmprints, histograms of differences **(HoD)** of orientation θ and (x, y) coordinates between minutiae of two palmprints can be used to shortlist strong candidates for forwarding to the matching process. Since palmprints produce a large number of minutiae, a reduction in the number of minutiae to be matched serves major computational benefits. Secondly, the selection of more reliable minutiae significantly improves matching accuracy. The false acceptance rate (FAR) is clearly reduced due to the elimination of most false minutiae. From the results, it can be concluded that the proposed algorithm is able to produce matching results comparable to state-of-the-art with much fewer minutiae matches. The proposed algorithm is independent of the dataset, minutiae encoding, and matching methods.

By conducting these studies, this thesis concludes that palmprint matching accuracy can be greatly improved by making the pre-processing of palmprints more efficient as the performance achieved in pre-processing dictates the performance achieved in matching. Instead of focusing only on the efficiency of matching algorithms, future research should lay equal emphasis on improving the performance of enhancement and feature selection methods for palmprints. Enhancement methods should cater to the unpredictable and abrupt changes in the palm ridge pattern

to extract reliable features. Secondly, future efforts should also focus on developing more intelligent algorithms to reduce the number of overall minutiae required for palmprint matching to improve computational efficiency.

8.2 Future Work

Seeing the effects of improved palmprint enhancement and feature selection, we intend to focus more on improving the pre-processing of palmprints. There is a need to perform a more elaborate comparison with other segmentation techniques and create masks that represent the palmar region in the image more sharply and help in further reducing feature search space. We are also interested in exploring the possibility of improving the accuracy of palmprint matching by coupling the proposed enhancement and feature selection methods with superior minutiae encoding schemes such as Minutiae Cylindrical Code and developing a robust minutiae matching algorithm that is both fast and efficient. We also intend to use GANs to reconstruct corrupted images prior to palmprint enhancement for better matching accuracy. Also, we intend to use the Graphical Processing Unit (GPU) to speed up the compute-intensive and time-consuming palmprint matching process. Lastly, we intend to create a custom palmprint dataset with artificially induced image degradations like poor contrast, occlusions, background texture, blurring, etc. to check the efficacy of the proposed methods on palmprints other than THUPALMLAB dataset.

Bibliography

[1] F. Galton, "Fingerprints (reprint)," *Da Capo Press*, vol. 63, pp. 4–7, 1965.

[2] "Fingerprint," 2015. [Online]. Available: https://en.wikipedia.org/wiki/Fingerprint

[3] R. S. Mohammed, N. J. Habeeb, and Z. M. Abood, "Iris matching using surf algorithm," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 12, pp. 91–102, 2016.

[4] "Minutia in nature," 2011. [Online]. Available: http://www.bromba.com/knowhow/MinutiaeStructures

[5] A. K. Jain, S. Prabhakar, and S. Pankanti, "On the similarity of identical twin fingerprints," *Pattern Recognition*, vol. 35, no. 11, pp. 2653–2663, 2002.

[6] A. S. for Surgery of the Hand *et al.*, *Essentials of hand surgery*. Lippincott Williams & Wilkins, 2002.

[7] A. K. Jain and J. Feng, "Latent palmprint matching," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 31, no. 6, pp. 1032–1047, 2009.

[8] L. Fei, G. Lu, W. Jia, S. Teng, and D. Zhang, "Feature extraction methods for palmprint recognition: A survey and evaluation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 346–363, 2018.

[9] H. Li, J. Qiu, and A. B. J. Teoh, "Palmprint template protection scheme based on randomized cuckoo hashing and minhash," *Multimedia Tools and Applications*, vol. 79, no. 17, pp. 11 947–11 971, 2020.

[10] A. Attia, S. Mazaa, Z. Akhtar, and Y. Chahir, "Deep learning-driven palmprint and finger knuckle pattern-based multimodal person recognition system," *Multimedia Tools and Applications*, vol. 81, no. 8, pp. 10 961–10 980, 2022.

[11] J. A., "Biometric authentication and identification market revenue worldwide in 2019 and 2027," https://www.statista.com/statistics/1012215/worldwide-biometric-authentication-and-identification-market-value/, 2022, accessed on 22.12.2022.

[12] "Palm recognition biometric market segmentation by type," https://www.researchnester.com/reports/palm-recognition-biometrics-market/2865, 2022, accessed on 23.12.2022.

[13] B. Liu and J. Feng, "Palmprint orientation field recovery via attention-based generative adversarial network," *Neurocomputing*, vol. 438, pp. 1–13, 2021.

[14] J. Khodadoust, M. A. Medina-Pérez, O. Loyola-González, R. Monroy, and A. M. Khodadoust, "A secure and robust indexing algorithm for distorted fingerprints and latent palmprints," *Expert Systems with Applications*, vol. 206, p. 117806, 2022.

[15] H. Fanchang, C. Xu, Y. Gongping, Y. Lu, L. Chengdong, L. Chenglong, and X. Chuanliang, "Local image quality measurement for multi-scale forensic palmprints," *Multimedia Tools and Applications*, vol. 79, no. 19, pp. 12 915–12 938, 2020.

[16] C. Zhou, J. Huang, F. Yang, and Y. Liu, "A hybrid fusion model of iris, palm vein and finger vein for multi-biometric recognition system," *Multimedia Tools and Applications*, vol. 79, no. 39, pp. 29 021–29 042, 2020.

[17] K. Bibi, S. Naz, and A. Rehman, "Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities," *Multimedia Tools and Applications*, vol. 79, no. 1, pp. 289–340, 2020.

[18] M. E. Rane and U. S. Bhadade, "Multimodal score level fusion for recognition using face and palmprint," *The International Journal of Electrical Engineering & Education*, p. 0020720920929662, 2020.

[19] D. D. Zhang, W. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Transactions on pattern analysis and machine intelligence*, 2003.

[20] T. S. Lee, "Image representation using 2d gabor wavelets," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 10, pp. 959–971, 1996.

[21] J. H. Van Deemter and J. H. du Buf, "Simultaneous detection of lines and edges using compound gabor filters," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 06, pp. 757–777, 2000.

[22] M. D. Bounneche, L. Boubchir, A. Bouridane, B. Nekhoul, and A. Ali-Chérif, "Multi-spectral palmprint recognition based on oriented multiscale log-gabor filters," *Neurocomputing*, vol. 205, pp. 274–286, 2016.

[23] J. Malik, D. Girdhar, R. Dahiya, and G. Sainarayanan, "Accuracy improvement in palmprint authentication system," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 4, p. 51, 2015.

[24] W. Jia, D.-S. Huang, and D. Zhang, "Palmprint verification based on robust line orientation code," *Pattern Recognition*, vol. 41, no. 5, pp. 1504–1513, 2008.

[25] D. Zhang, W. Zuo, and F. Yue, "A comparative study of palmprint recognition algorithms," *ACM computing surveys (CSUR)*, vol. 44, no. 1, p. 2, 2012.

[26] "Polyu palmprint database (version 2.0),multispectral palmprint database." 2008. [Online]. Available: http://www.comp.polyu.edu.hk/biometrics/

[27] A.-K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1. IEEE, 2004, pp. 520–523.

[28] L. Fei, Y. Xu, W. Tang, and D. Zhang, "Double-orientation code and nonlinear matching scheme for palmprint recognition," *Pattern recognition*, vol. 49, pp. 89–101, 2016.

[29] Z. Guo, D. Zhang, L. Zhang, and W. Zuo, "Palmprint verification using binary orientation co-occurrence vector," *Pattern Recognition Letters*, vol. 30, no. 13, pp. 1219–1227, 2009.

[30] L. Zhang, H. Li, and J. Niu, "Fragile bits in palmprint recognition," *IEEE Signal processing letters*, vol. 19, no. 10, pp. 663–666, 2012.

[31] L. Fei, B. Zhang, Y. Xu, and L. Yan, "Palmprint recognition using neighboring direction indicator," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 6, pp. 787–798, 2016.

[32] L. Fei, J. Wen, Z. Zhang, K. Yan, and Z. Zhong, "Local multiple directional pattern of palmprint image," *arXiv preprint arXiv:1607.06166*, 2016.

[33] S. Zhao and B. Zhang, "Joint constrained least-square regression with deep convolutional feature for palmprint recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 511–522, 2020.

[34] "Iitd touchless palmprint database (version 1.0)." 2006. [Online]. Available: http://www4.comp.polyu.edu.hk/

[35] "Gpds palmprint image database." 2011. [Online]. Available: http://www.gpds.ulpgc.es

[36] "Casia palmprint image database." 2005. [Online]. Available: http://biometrics.idealtest.org/

[37] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[38] X. Wu, Q. Zhao, and W. Bu, "A sift-based contactless palmprint verification approach using iterative ransac and local palmprint descriptors," *Pattern Recognition*, vol. 47, no. 10, pp. 3314–3326, 2014.

[39] A. Kong, D. Zhang, and M. Kamel, "Palmprint identification using feature-level fusion," *Pattern Recognition*, vol. 39, no. 3, pp. 478–487, 2006.

[40] G. K. O. Michael, T. Connie, and A. B. J. Teoh, "Touch-less palm print biometrics: Novel design and implementation," *Image and Vision Computing*, vol. 26, no. 12, pp. 1551–1560, 2008.

[41] Y.-T. Luo, L.-Y. Zhao, B. Zhang, W. Jia, F. Xue, J.-T. Lu, Y.-H. Zhu, and B.-Q. Xu, "Local line directional pattern for palmprint recognition," *Pattern Recognition*, vol. 50, pp. 26–44, 2016.

[42] F. Zhong and J. Zhang, "Face recognition with enhanced local directional patterns," *Neurocomputing*, vol. 119, pp. 375–384, 2013.

[43] Q. Zhao, W. Bu, and X. Wu, "Sift-based image alignment for contactless palmprint verification," in *Biometrics (ICB), 2013 International Conference on*. IEEE, 2013, pp. 1–6.

[44] "Thu 500ppi palmprint database," 2012. [Online]. Available: http://ivg.au.tsinghua.edu.cn/

[45] D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 1, pp. 27–40, 1997.

[46] A. M. Bazen and S. H. Gerez, "Segmentation of fingerprint images," in *ProRISC 2001 Workshop on Circuits, Systems and Signal Processing*. Veldhoven, the Netherlands, 2001, pp. 276–280.

[47] S. A. Tariq, S. Iqbal, M. Ghafoor, I. A. Taj, N. M. Jafri, S. Razzaq, and T. Zia, "Massively parallel palmprint identification system using gpu," *Cluster Computing*, vol. 22, no. 3, pp. 7201–7216, 2019.

[48] M. Ghafoor, S. A. Tariq, I. A. Taj, N. M. Jafri, and T. Zia, "Robust palmprint identification using efficient enhancement and two-stage matching technique," *IET Image Processing*, vol. 14, no. 11, pp. 2333–2342, 2020.

[49] B.-G. Kim, H.-J. Kim, and D.-J. Park, "New enhancement algorithm for fingerprint images," in *2002 International Conference on Pattern Recognition*, vol. 3. IEEE, 2002, pp. 879–882.

[50] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, pp. 777–789, 1998.

[51] M. Ghafoor, I. A. Taj, W. Ahmad, and N. M. Jafri, "Efficient 2-fold contextual filtering approach for fingerprint enhancement," *IET Image Processing*, vol. 8, no. 7, pp. 417–425, 2014.

[52] S. Chikkerur, A. N. Cartwright, and V. Govindaraju, "Fingerprint enhancement using stft analysis," *Pattern recognition*, vol. 40, no. 1, pp. 198–211, 2007.

[53] M. Ghafoor, I. A. Taj, and M. N. Jafri, "Fingerprint frequency normalisation and enhancement using two-dimensional short-time fourier transform analysis," *IET Computer Vision*, vol. 10, no. 8, pp. 806–816, 2016.

[54] R. Cappelli, M. Ferrara, and D. Maio, "A fast and accurate palmprint recognition system based on minutiae," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 956–962, 2012.

[55] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[56] J. Dai and J. Zhou, "Multifeature-based high-resolution palmprint recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 945–957, 2010.

[57] H. Soleimani and M. Ahmadi, "Fast and efficient minutia-based palmprint matching," *IET Biometrics*, vol. 7, no. 6, pp. 573–580, 2018.

[58] I. S. Hussein, S. B. Sahibuddin, M. J. Nordin, and N. N. B. A. Sjarif, "Multimodal recognition system based on high-resolution palmprints," *IEEE Access*, vol. 8, pp. 56 113–56 123, 2020.

[59] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 4, pp. 348–359, 1999.

[60] M. I. Ahmad, W. L. Woo, and S. Dlay, "Non-stationary feature fusion of face and palmprint multimodal biometrics," *Neurocomputing*, vol. 177, pp. 49–61, 2016.

[61] J. Dai, J. Feng, and J. Zhou, "Robust and efficient ridge-based palmprint matching," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 8, pp. 1618–1632, 2011.

[62] M. Tico and P. Kuosmanen, "An algorithm for fingerprint image postprocessing," in *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No. 00CH37154)*, vol. 2. IEEE, 2000, pp. 1735–1739.

[63] N. K. Ratha, S. Chen, and A. K. Jain, "Adaptive flow orientation-based feature extraction in fingerprint images," *Pattern recognition*, vol. 28, no. 11, pp. 1657–1672, 1995.

[64] D. D. Hung, "Enhancement and feature purification of fingerprint images," *Pattern Recognition*, vol. 26, no. 11, pp. 1661–1671, 1993.

[65] Q. Xiao and H. Raafat, "Fingerprint image postprocessing: a combined statistical and structural approach," *Pattern Recognition*, vol. 24, no. 10, pp. 985–992, 1991.

[66] A. M. Bazen, G. T. Verwaaijen, S. H. Gerez, L. P. Veelenturf, and B. J. Van Der Zwaag, "A correlation-based fingerprint verification system," in *Proceedings of the ProRISC2000 workshop on circuits, systems and signal processing*, 2000, pp. 205–213.

[67] M. Tico and P. Kuosmanen, "Fingerprint matching using an orientation-based minutia descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1009–1014, 2003.

[68] A. Jain, L. Hong, and R. Bolle, "On-line fingerprint verification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 4, pp. 302–314, 1997.

[69] X. Jiang and W.-Y. Yau, "Fingerprint minutiae matching based on the local and global structures," in *Proceedings 15th international conference on pattern recognition. ICPR-2000*, vol. 2. IEEE, 2000, pp. 1038–1041.

[70] X. Chen, J. Tian, and X. Yang, "A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure," *IEEE Transactions on image processing*, vol. 15, no. 3, pp. 767–776, 2006.

[71] X. Jiang and W.-Y. Yau, "Fingerprint minutiae matching based on the local and global structures," in *Proceedings 15th international conference on pattern recognition. ICPR-2000*, vol. 2. IEEE, 2000, pp. 1038–1041.

[72] W. Jia, J. Gao, W. Xia, Y. Zhao, H. Min, and J.-T. Lu, "A performance evaluation of classic convolutional neural networks for 2d and 3d palmprint and palm vein recognition," *International Journal of Automation and Computing*, vol. 18, no. 1, pp. 18–44, 2021.

[73] Y. Liu and A. Kumar, "Contactless palmprint identification using deeply learned residual features," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 2, pp. 172–181, 2020.

[74] S. A. Kumar, G. C. P. Latha, and O. Hemavathy, "Palm print identification and classification in the field of biometric in cnn using knn," in *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. IEEE, 2022, pp. 1–4.

[75] E. A. Alrahawe, V. T. Humbe, and G. Shinde, "A contactless palmprint biometric system based on cnn," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 13, pp. 6344–6356, 2021.

[76] K. Ito, Y. Suzuki, H. Kawai, T. Aoki, M. Fujio, Y. Kaga, and K. Takahashi, "Handsegnet: Hand segmentation using convolutional neural network for

contactless palmprint recognition," *IET Biometrics*, vol. 11, no. 2, pp. 109–123, 2022.

[77] J. P. Veigas and S. Kumari M, "Deep learning approach for touchless palmprint recognition based on alexnet and fuzzy support vector machine," *International journal of electrical and computer engineering systems*, vol. 13, no. 7, pp. 551–559, 2022.

[78] Q. Zhou, W. Jia, and Y. Yu, "Multi-stream convolutional neural networks fusion for palmprint recognition," in *Biometric Recognition: 16th Chinese Conference, CCBR 2022, Beijing, China, November 11–13, 2022, Proceedings*. Springer, 2022, pp. 72–81.

[79] M. Ahmadi and H. Soleimani, "Palmprint image registration using convolutional neural networks and hough transform," *arXiv preprint arXiv:1904.00579*, 2019.

[80] J.-i. Funada, N. Ohta, M. Mizoguchi, T. Temma, K. Nakanishi, A. Murai, T. Sugiuchi, T. Wakabayashi, and Y. Yamada, "Feature extraction method for palmprint considering elimination of creases," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, vol. 2. IEEE, 1998, pp. 1849–1854.

[81] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[82] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

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

[84] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[85] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," *arXiv preprint arXiv:1606.08921*, 2016.

[86] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," *Advances in neural information processing systems*, vol. 29, 2016.

[87] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[88] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[89] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[90] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[91] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[92] C. L. Wilson, G. T. Candela, C. I. Watson *et al.*, "Neural network fingerprint classification," *Journal of Artificial Neural Networks*, vol. 1, no. 2, pp. 203–228, 1994.

[93] X. Liu, W. Liang, Y. Wang, S. Li, and M. Pei, "3d head pose estimation with convolutional neural network trained on synthetic images," in *2016*

IEEE international conference on image processing (ICIP). IEEE, 2016, pp. 1289–1293.

[94] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[95] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1513–1520.

[96] C.-R. Chou, B. Frederick, G. Mageras, S. Chang, and S. Pizer, "2d/3d image registration using regression learning," *Computer Vision and Image Understanding*, vol. 117, no. 9, pp. 1095–1106, 2013.

[97] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[98] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[99] T. Tieleman, G. Hinton *et al.*, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[100] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[101] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[102] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[103] S. Chikkerur, A. Cartwright, and V. Govindaraju, "K-plet and coupled bfs: a graph based fingerprint representation and matching algorithm," *Advances in biometrics*, pp. 309–315, 2005.

[104] R. Cappelli, M. Ferrara, and D. Maio, "A fast and accurate palmprint recognition system based on minutiae," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 956–962, 2012.