A Novel Convolutional Neural Network Pore-Based Fingerprint Recognition System

Mohammed Ali

A Thesis

In the Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

March 2024

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Mohammed Ali

Entitled: A Novel Convolutional Neural Network Pore-Based Fingerprint Recognition System

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____Chair
Dr. S. Samuel Li

____Examiner External
Dr. Wasfy B. Mikhael

____Arm’s Length Examiner
Dr. Chun-Yi Su

____Examiner
Dr. M.N.S. Swamy

____Examiner
Dr. Wei-Ping Zhu

____Thesis Supervisor
Dr. M. Omair Ahmad

____Thesis Supervisor
Dr. Chunyan Wang

Approved by ____
Dr. Jun Cai, Graduate Program Director

April 26, 2024 ____
Dr. Mourad Debbabi, Dean, Gina Cody School of Engineering and Computer Science

Abstract

A Novel Convolutional Neural Network Pore-Based Fingerprint Recognition System

Mohammed Ali, Ph.D.

Concordia University, 2024

Biometrics play an important role in security measures, such as border control and online transactions, relying on traits like uniqueness and permanence. Among the different biometrics, the fingerprint stands out for their enduring nature and individual uniqueness. Fingerprint recognition systems traditionally rely on ridge patterns (Level 1) and minutiae (Level 2). However, these systems suffer from recognition accuracy with partial fingerprints. Level 3 features, such as pores, offer distinctive attributes crucial for individual identification, particularly with high-resolution acquisition devices. Moreover, the use of convolutional neural networks (CNNs) has significantly improved the accuracy in automatic feature extraction for biometric recognition.

A CNN-based pore fingerprint recognition system consists of two main modules, pore detection and pore feature extraction and matching modules. The first module generates pixel intensity maps to determine the pore centroids, while the second module extracts relevant features of pores to generate pore representations for matching between query and template fingerprints. However, existing CNN architectures lack in generating deep-level discriminative feature and computational efficiency. Moreover, available knowledge on the pores has not been taken into consideration optimally for pore centroids and metrics other than Euclidean distance have not been explored for pore matching.

The objective of this research is to develop a CNN-based pore fingerprint recognition scheme that is capable of providing a low-complexity and high-accuracy performance. The design of the CNN architecture of the two modules aimed at generating features at different hierarchical levels in residual frameworks and fusing them to produce comprehensive sets of discriminative features. Depthwise and depthwise separable convolution operations are judiciously used to keep the complexity of networks low. In the proposed pore centroid part, the knowledge of the variation of the pore characteristics is used. In the proposed pore matching scheme, a composite metric, encompassing the Euclidean distance, angle, and magnitudes difference between the vectors of pore representations, is proposed to measure the similarity between the pores in the query and template images.

Extensive experiments are performed on fingerprint images from the benchmark PolyU High-Resolution-Fingerprint dataset to demonstrate the effectiveness of the various strategies developed and used in the proposed scheme for fingerprint recognition.

Acknowledgment

First and foremost, I extend my deepest gratitude to Allah Almighty for the unwavering strength and guidance provided throughout this endeavor. I would like to express my heartfelt gratitude to several individuals whose support and encouragement have been instrumental in the completion of this thesis.

I extend my personal thanks to my two supervisors, Dr. M. O. Ahmad and Dr. Chunyan Wang, for their invaluable guidance, unwavering patience, and scholarly insights throughout this research journey. Their expertise, constructive feedback, and constant encouragement have been invaluable in shaping this work.

I am profoundly grateful to my parents and siblings for their unwavering love, support, and understanding. Their belief in me has been a constant source of motivation, and I am indebted to them for their sacrifices and encouragement.

I dedicate this thesis to the memory of my dear sister Sara, whose untimely passing deeply impacted me. Though she is no longer with us, her memory continues to inspire me to strive for excellence and pursue my academic endeavors with diligence and passion.

To my beloved wife Malaz, your unwavering love, patience, and understanding have been my rock throughout this challenging journey. Your sacrifices and encouragement have sustained me during the highs and lows of academic life, and I am profoundly grateful for your unwavering support.

Lastly, I extend my appreciation to my friends for their encouragement, camaraderie, and moral support. Your presence in my life has made this journey more fulfilling, and I am thankful for the moments of laughter and companionship shared along the way.

CONTENTS

List of Figures

List of Tables

List of Symbols

α_k, β_k	Weights used for the loss function $Loss_{id_k}$ and $Loss_{vr_k}$, respectively
β	Niblack factor
β_l	Locally adaptive Niblack factor
δ	Threshold for similarity measure
θ	Angle between two vectors
θ'	Adjusted Angle
σ_g	Global standard deviation
σ_l	Local standard deviation
γ	Pore intensity value
b_j^l	Trainable bias parameter
C	Set of pore correspondence
d_{min}	Minimum distance between the pores coordinates
$f(\,)$	Convolution function
L	Binary cross-entropy loss function
$Loss_{id_k}$	Identification loss functions at the kth stage
$Loss_{vr_k}$	Verification loss functions at the kth stage
$Loss_{id}$	Identification loss function
$Loss_{total}$	Total loss function
$Loss_{vr}$	Verification loss function
ɱ	Mean deviation of the pore intensities
б	Standard deviation of the pore intensities

m_g	Global mean
m_l	Local mean
N	Window size in the binarization step
n	Number of filters in a convolution layer
N_B	Batch size
N_l	Number of pixels in the l^{th} window
N_p	Number of pixels within the pore region of the candidate pore
p	Patch size
$\widehat{pb}_i$	Predicted probability distribution
pb_i	Target probability distribution
Pc_{k}	Pore centroid
PF_{Q_i}	Feature representation of the i$^{\mathrm{th}}$ pore in the query image
PF_{T_i}	Feature representation of the i$^{\mathrm{th}}$ pore in the template image
p_k	Intensity value of the k^{th} pixel in a pore intensity map
P_{Qi}	i$^{\mathrm{th}}$ pore in the query fingerprint image
P_{Ti}	i$^{\mathrm{th}}$ pore in the template fingerprint image
RC	Refined pore correspondence list
s	Convolution filter size
S_{ij}	Similarity matrix
t	Maximum intensity threshold required for true pore
T_{CNN}	Average time per image for the neural network part of pore detection module
T_l	Spatially adaptive threshold

T_P	Average time per image for the postprocessing part of pore detection module
T_T	Total average time per image for the pore detection module
W_{ij}^{l}	Trainable filter (kernel)
$X_{d_{min}}$	Refined pore map
X_o	Pore intensity map
X_t	Binary map

List of Abbreviation

AFRS	Automated fingerprint recognition system
ANN	Artificial neural network
AP	Alignment-based pore matching
APFS	Automatic partial fingerprint system
BN	Batch normalization
CAGR	Compound annual growth rate
CNN	Convolutional neural network
CPU	Central processing unit
DAPM	Dynamic anisotropic pore model
DB	Database
DNA	Deoxyribonucleic acid
DOG	Difference of Gaussian
DP	Direct pore matching
DPI	Dots per inch
ED	Euclidean distance
EER	Equal error rate
FBI	Federal Bureau of Investigation
FCN	Fully connected network
FLOPs	Floating-point multiplication-addition operations
FMR	False matching rate
FN	False negative
FNMR	False negative matching rate
FP	False positive
GB	Giga bite
GHz	Giga hertz
GP	Global pooling
GPU	Graphics processing unit
HBM	High bandwidth memory

HRF	High-resolution-fingerprint
ICP	Iterative closest point
ID	Identity
LDP	Location determination problem
LTPM	Local topology-preserving matching
MD	Magnitude difference
MICP	Modified iterative closest point
MSCOCO	Microsoft common objectives in context
PC	Personal computer
PolyU	Hong Kong polytechnic university
RAM	Random access memory
RANSAC	Random sample consensus
ReLU	Rectified linear unit
ROC	Receiver operating characteristic
RootSIFT	Root scale-invariant feature transform
SDG	Stochastic gradient descent
SS	Sector similarity
STD	Standard deviation
TN	True negative
TP	True positive
TS	Triangle similarity
TS_SS	Triangle similarity and sector similarity
WRANSAC	Weighted random sample consensus

Chapter 1

Introduction

1.1 General

In the current landscape characterized by the pervasive integration of technology into various aspects of modern living, the safeguarding of confidential information has become an increasingly complex endeavor. With the rapid pace of digitization, traditional means of ensuring data security, such as passwords and keys [1], have gradually revealed their limitations and susceptibility to sophisticated hacker attacks. The exponential growth in cyber threats has underscored the inherent vulnerability of passwords, rendering them the weakest link in an organization's security infrastructure [2].

Recent incidents of network security breaches and identity thefts have served as alarming wake-up calls, shedding light on the urgent need for a robust authentication method that can effectively combat the evolving landscape of cyber threats. This heightened awareness has prompted a significant shift in focus toward the realm of biometric security, which offers a promising solution for the pressing challenge of verifying individual identities with utmost accuracy and reliability [2].

Biometric security leverages the distinctive physical and behavioral traits unique to each individual, including but not limited to fingerprints, face, iris, gait, and voice, as shown in Fig. 1.1. These biometric characteristics possess an inherent and indelible connection to an individual's very being, making them an ideal foundation for robust authentication mechanisms. Unlike passwords that can be easily shared or compromised, biometric traits are intrinsically linked to an individual and cannot be easily replicated or imitated [3], [4].

By harnessing the power of biometric security systems, organizations can establish a highly accurate and reliable means of confirming an individual's identity. Biometric authentication surpasses the limitations of traditional methods, providing a formidable defense against the vulnerabilities inherent in password-based systems. Through the utilization of biometric traits, organizations can achieve unparalleled precision in identity verification, effectively mitigating the risks associated with unauthorized access and identity fraud [2].

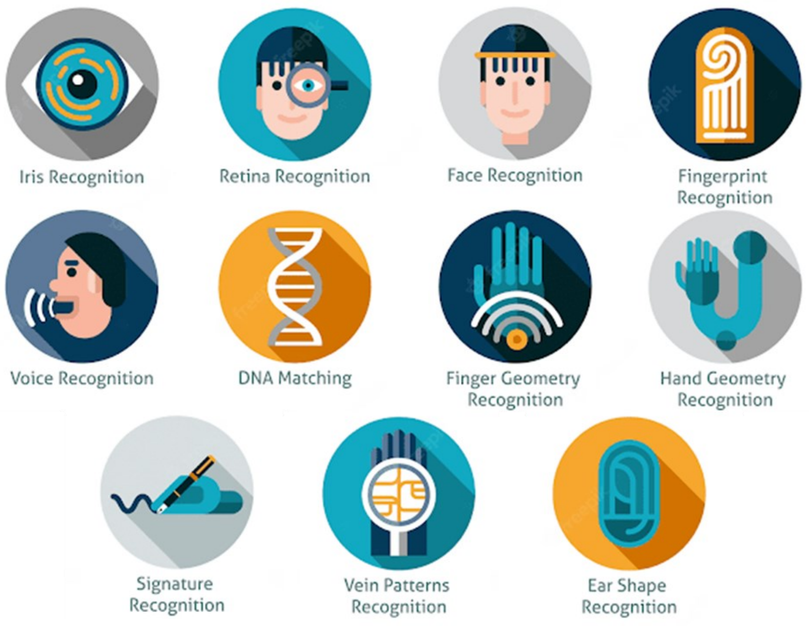


Figure 1.1: Examples of biometric authentication.

As the realm of biometric security continues to evolve and mature, advancements in technology are continuously enhancing the effectiveness and reliability of these authentication systems. Ongoing research and development efforts are dedicated to refining biometric algorithms, improving sensor capabilities, and addressing concerns related to privacy and ethical considerations [2], [5], [6].

Biometric security systems have emerged as a compelling alternative, promising enhanced protection against hacking attempts and rendering replication or theft virtually impossible. By leveraging distinctive physical or behavioral characteristics, biometrics offer a seamless and convenient user experience by obviating the need for repetitive password inputs across multiple devices.

One notable advantage of biometric authentication lies in the arduous task of falsifying biometric modalities. These unique traits, inherent to each individual, exhibit a remarkable stability throughout one's lifetime, further solidifying their efficacy in ensuring robust identification processes. As organizations integrate biometric security systems, employees stand to benefit from the convenience and heightened levels of protection bestowed upon their computer systems and sensitive files. Liberated from the burdensome task of remembering complex passwords or frequently changing them for security purposes, individuals can allocate their valuable time and resources toward productive endeavors, instead of grappling with password resets.

Unlike their password counterparts, biometric traits possess inherent non-transferability, drastically reducing time-consuming authentication procedures. With average identification times typically under 5 seconds, the efficiency and swiftness of biometric authentication are unparalleled. Moreover, the immutable nature of biometric identifiers effectively eliminates any potential for individuals to deny their actions, as these unique traits are an indelible part of their very being. Attempts to duplicate biometric characteristics are met with insurmountable challenges, further fortifying the system's integrity.

The pursuit of reliable authentication mechanisms has long captivated the attention of researchers and industry practitioners, a pursuit that assumes heightened significance in an era of escalating security concerns. Biometrics, characterized by the automated identification of individuals through their distinctive physical or behavioral characteristics, have emerged as potent tools for achieving this objective [7]. These pattern recognition systems play a pivotal role in verifying the authenticity of specific biometric features possessed by users, thereby establishing their identity.

The universal and unique attributes of biometric traits have propelled them to the forefront of human identification methodologies [8]. As a result, novel techniques and methodologies have emerged, capitalizing on these distinct biometric characteristics to facilitate user identity recognition and verification processes [9]. The continual advancement of biometric technologies, coupled with ongoing research and development efforts, holds great promise in further improving the accuracy, reliability, and scalability of biometric security systems.

Biometric-based authentication systems have been extensively studied in the literature, with biometrics such as iris, voice, fingerprints, and facial characteristics being among the most commonly used. Among these, fingerprints have been widely accepted and used for a long time. In fact, the Chinese have been using fingerprints to sign documents for over a millennium, which is a testament to the longevity and reliability of this biometric modality [10].

According to a report by BCC Research [11], the global biometric market was valued at \$21.6 billion in 2021 and is projected to grow at a compound annual growth rate (CAGR) of 15.7% during the forecast period (2021 - 2026), reaching a value of \$44.7 billion by 2026. The market for Automatic Fingerprint Recognition System (AFRS) and fingerprint biometric technologies accounted for the largest share of the global biometrics market and was projected to continue to be the primary source of overall market revenues from 2021 to 2026 and beyond. The main user segments of biometric technology include government departments, law enforcement, military, and transport and aviation markets. These statistics demonstrate the widespread use of fingerprint biometrics.

Fingerprint properties are highly accurate and unique to each individual. Authentication systems based on fingerprints have been shown to produce low false acceptance and false rejection rates, along with other benefits such as an easy and cost-effective implementation procedure. Additionally, fingerprints typically remain unchanged from birth until death [12]. In addition to being unique and unchanging, fingerprints can be collected in a non-invasive manner with no side effects [13].

Fingerprint recognition technology has advanced significantly in recent years. While full prints can be matched with high accuracy, partial fingerprint recognition still requires improvement. The need for partial fingerprint recognition systems is increasing in both civilian and forensic applications. In civilian applications, small hand-held devices such as mobile phones and miniaturized fingerprint sensors produce only small portions of fingerprints. However, the miniaturization of fingerprint sensors has led to small sensing areas of 0.42" x 0.42". According to FBI specifications, an average of 1.0" x 1.0" fingerprint area is needed to collect minutiae points accurately (minimum 12 minutiae points).

The accuracy of recognition while using partial fingerprints is reduced due to the reduced fingerprint size, and research is still ongoing to improve this. The development of an automatic partial fingerprint recognition system (APFS) that can achieve high accuracy, low error rate, and maximum speed is still a work in progress. The present research aims to develop an APFS that can authenticate an individual.

1.2 Biometric Technology

Biometrics is a field of study that deals with verifying and establishing the identity of an individual through their physiological features or behavioral traits. Biometric technologies come in different levels of complexity, capabilities, and performance, but they all share some common elements. Biometric identification systems are essentially pattern recognition systems that use acquisition devices such as cameras and scanning devices to capture images, recordings, or measurements of an individual’s characteristics. Computer hardware and software are then used to extract, encode, store, and compare these characteristics. Since the process is automated, biometric decision-making is generally very fast and takes only a few seconds in real-time.

Biometric systems can operate in two modes: verification and identification [4]. Verification, or authentication, confirms a person’s identity by checking if they are who they claim to be. Identification, on the other hand, determines a person’s identity by finding out who they are. Biometric systems use various methods to measure different features, but they all share similar processes that consist of two main stages: (I) Enrollment and (II) Verification or Identification.

The enrollment process teaches a biometric system to recognize a specific person. The person first shows an identifier, such as an ID card. The biometric matches the identity on the ID document. Then, the person gives the biometric (e.g., fingerprint or iris) to a capture device. The unique features are found, and one or more samples are extracted, encoded and stored as a reference template for future comparisons. Templates are usually stored remotely in a central database or in the biometric reader device itself. The quality of the template is vital for the biometric application to work well [14]. Verification systems check if a person is who they say they are after enrollment. The person gives an identifier and a biometric, which the system captures and compares with their stored template. Identification systems find out who a person is after enrollment. The person only gives a biometric, which the system compares with all the stored templates. The verification systems can match a biometric with one template in less than a second while the identification systems can match a biometric with many templates.

Biometric systems capture a template that is likely to be different each time. They use a threshold to decide how similar the trial and reference templates should be. Biometric systems can make a match or no-match decision based on this threshold. They generate a score that shows how similar the templates are and compare it to the threshold. In identification systems, the threshold can allow more than one reference template to match the trial template, with the highest scores being the closest matches.

1.2.1 Types of Biometric Technologies

Biometric technologies have been increasingly developed and deployed for many years [15]. Based on the human characteristics, biometrics fall into physiological or behavioral category [16]. Physiological features, such as fingerprint, face, DNA, ear, palm print, hand geometry, iris and retina, are related to the shape of the body. Behavioral features, such as signature, keystroke dynamics, gait and voice, are related to the actions of a person. The latter type of biometrics is sometimes referred to as behaviometrics. The choice of a biometric trait depends on the requirements of the application, as each trait has its advantages and disadvantages [17]. Fig. 1.1 shows some of the commonly used biometric systems.

Facial recognition is a biometric technology that identifies people by analyzing their facial features, such as the eyes, eyebrows, nose, lips and chin [18]. However, face recognition faces challenges such as varying views and lighting conditions, as well as the aging of the face. These factors make it difficult to determine if the face alone is sufficient to recognize a person among many identities.

Deoxyribonucleic acid (DNA) molecules is a long-time storage of information [19]. DNA profiling is a process that uses DNA from blood, semen, skin, saliva or hair at a crime scene to find a match with an individual. This method is very accurate for identifying matching DNA. However, it is extremely expensive and requires long process to identify an individual.

Hand geometry is a biometric technology that uses 96 measurements of the hand, such as the width, height, and length of the fingers, distances between joints and shapes of the knuckles, to identify individuals [20]. It is a simple, easy and inexpensive method that has been widely used in various locations. It is not affected by many environmental or anomaly factors and has a high accuracy. However, it has some limitations, such as the variability of the hand geometry over time, the inability to recognize a large number of people, and the large size of the scanner required. Therefore, it is not widely adopted for computer security applications.

Signature recognition is a biometric technology that verifies the writer’s identity by matching the signature with samples stored in a database [21]. It analyzes the unique way a person signs his name, either by looking at how the signature was written or by estimating how the signature was formed. However, this type of biometric is not stable over time, as the same person can sign differently depending on physical or emotional factors.

Iris recognition is a biometric technology that identifies people by analyzing the features of the colored tissue around the pupil, such as rings, rows and spots [22]. The iris has a very intricate and unique texture that makes it ideal for recognition purposes. Even the irises of identical twins are not the same. Although this complexity and uniqueness enable the system to be more precise and suitable for large-scale identification systems, iris recognition is expensive and requires advanced technology to get high accuracy.

Keystroke dynamics is a biometric technology that identifies people by analyzing how they type on a computer keyboard. It is based on the idea that human repetitive actions are predictable and unique to each individual [23]. It considers factors such as the time between key presses and releases, the type of keyboard used, and the emotional and physical state of the person. It does not require any special hardware, only the regular computer keyboard. It is not as distinctive as other biometrics, but it can be adequate for some applications.

Gait recognition is a behavioral biometric modality that uses the unique walking pattern of a person to identify or authenticate them from a distance. A gait recognition system employs a standard camera to capture the video of a person walking and applies computer vision algorithms to extract the silhouette and the movement features of the person. The system can then track and match the person with a database of known gait models. However, gait recognition is not very robust, as it can be influenced by various factors such as clothing, footwear, terrain, fatigue, and health. Gait is also not very distinctive across individuals, making it unsuitable for high-security applications [24].

Speech or voice recognition is a biometric modality that uses the unique features of a person's voice to identify or authenticate them [25]. Voice features are determined by both physical and behavioral factors. Physical factors are related to the anatomy of the vocal organs, such as the mouth, lips, vocal cords, and nasal passages. Behavioral factors are influenced by the mood, health, and age of the person. The physical factors tend to be stable, while the behavioral factors can vary over time. Because of this variability, voice recognition is not a very reliable biometric modality.

Fingerprint recognition is a biometric technology that identifies or authenticates a person based on their unique fingerprint [4]. This is the core topic of this research, and it will be explained in depth in the next sections. Fingerprint recognition is one of the most widely used biometric modalities, and it has been commercially available since the early 1970s [26].

Despite the apparent advantages of biometric systems for reliable person authentication, they are not infallible methods of automatic human recognition. The study of the current methods, along with the emergence of cheap, small biometric sensors and fast processing chips, reveals that a wider application of biometric technology would demand improved solutions to address three essential challenges [27]: system security, privacy issues, and recognition performance.

To achieve system security (that is, to prevent fraudsters from accessing the system) and privacy issues (that is, to prevent unauthorized system administrators from exploiting the system), the biometric modality selected should have the following properties:

- The physical trait should be stable and consistent throughout the person's life.
- The physical trait should be unique and specific to the individual person.
- The data should be easily verified against the actual person in a simple, automated manner.

These properties are fully met by the fingerprint biometric. Fingerprint recognition is one of the oldest and most trusted biometric modalities. Fingerprint recognition is based on the premise that every person has a unique and unchangeable fingerprint, even identical twins. This makes fingerprint recognition suitable for many applications that require identity verification, such as forensics, access control, website, and device security. Furthermore, the hardware and software for fingerprint recognition are becoming more affordable and accessible.

1.2.2 Fingerprint Biometric

A fingerprint is a mark left by the friction ridges on the surface of a human finger tip. A friction ridge is a raised part of the skin on the fingers that consists of one or more connected units of ridge skin (see Fig. 1.2). These are also called "epidermal ridges", and they are formed by the interaction between the dermal papillae of the lower skin layer and the interpapillary pegs of the upper skin layer. These epidermal ridges enhance the vibrations caused by, for example, touching an uneven surface, and improve the transmission of the signals to the sensory nerves that perceive fine texture [28]. These ridges also help in holding rough or wet surfaces. Fingerprint impressions can be left on a surface by the natural sweat from the eccrine glands in the friction ridge skin, or they can be made by ink or other substances that transfer from the ridge peaks to a smooth surface, such as a fingerprint card [29]. Fingerprint records usually include impressions from the pad of the last finger joint, as well as some parts of the lower finger joint areas.

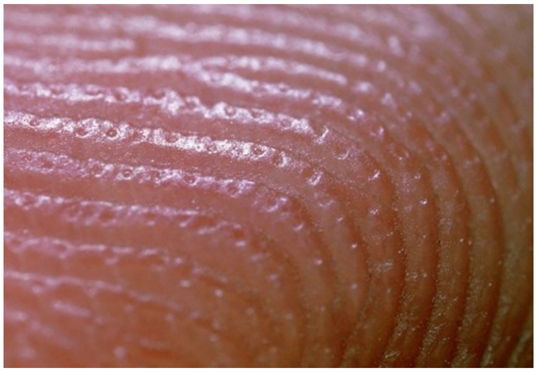


Figure 1.2: A macro photo of a human fingerprint showing the unique pattern of ridges and valleys.

The ridge patterns on our fingers are unique and never repeated, making fingerprints an ideal biometric trait for authentication purposes. These patterns develop during the fetal stage and remain constant throughout life. Fingerprint-based biometric systems are used by more than 74% of the biometric market as an authentication tool [4].

Fingerprints are a universally accepted solution for authentication, with over 96% of the population having legible fingerprints, which is more than the number of people who have licenses and identity cards. They are highly distinctive and one of the most accurate forms of biometrics available. In addition to these advantages, the following benefits have also been recognized [30]:

i. The acquisition process is non-intrusive and requires no training.
ii. It is the most economical biometric PC user authentication technique.
iii. Small storage space is required for the biometric template, reducing the size of the database memory required.

1.3 Basic Principles of Fingerprint Technology

Sir Francis Galton, a British scientist, is regarded as the Father of Fingerprint Science. He established the basic principles of fingerprint technology [31], which are as follows:

- Consistency in Persistence: Fingerprints retain their unchanging nature from birth until decomposition occurs. The ridges on the friction skin of fingers develop distinct and permanent characteristics within one and a half to two months of pregnancy, reaching full formation in 180 days. These formations grow proportionally with the body, ensuring permanence unless the skin deteriorates or sustains damage.

- Uniqueness and Diversity: No two fingerprints are identical, except those taken from the same finger of the same individual. Even fingerprints captured from different fingers of the same person exhibit distinct patterns. Hence, fingerprints serve as a unique, divine seal for individual identification.
- Immutability of Fingerprints: As previously mentioned, the human skin comprises two layers, the outer epidermis and the inner dermis. Temporary effects on the ridges in the epidermis, such as simple injuries, aging, growth, skin peeling, warts, creases, burns, or skin diseases, may occur. However, over time, these affected ridges revert to their original pattern and characteristics. On the other hand, scars resulting from skin wounds affecting the dermis are permanent and alter the formation of patterns and ridges.

Therefore, it can be comprehended that fingerprints remain flawless throughout a person's entire life, making them the most dependable, truthful, immutable, and steadfast means of establishing one's identity.

1.3.1 Practical Applications and Usage

Fingerprint biometric finds extensive applications in various fields, including entrance control, door-lock systems, smart cards, vehicle ignition control, and fingerprint-based access. With advancements in technology leading to smaller fingerprint sensor sizes, the scope of applications has expanded to the mobile market. Given the rapid growth of the current mobile market, its potential surpasses that of all other application markets. The fingerprint markets are categorized as illustrated in Fig. 1.3.

In criminology, fingerprints are a frequently employed and highly reliable source for identifying criminals and individuals involved in fraudulent activities. Their utility extends to the identification of unidentified bodies, where fingerprint records are compared with existing databases. Fingerprint technology is increasingly integrated into crucial service-oriented industries such as banking, licensing, and passport issuance. The use of fingerprints significantly reduces the occurrences of forgery, impersonation, and fraud in these sectors. Furthermore, fingerprints play a vital role in property and civil cases, leveraging historical records from registration departments and documents to resolve critical issues.

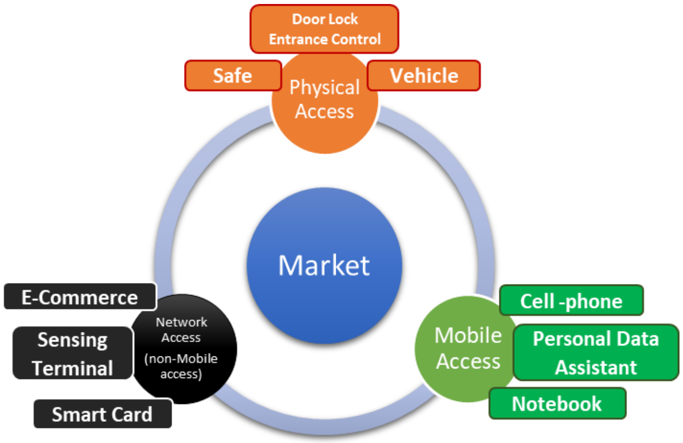


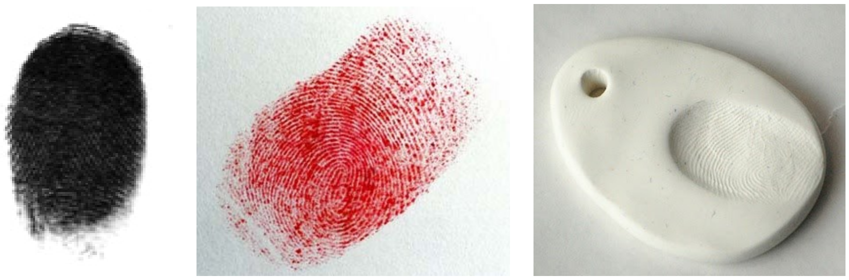
Figure 1.3: Application Areas of Fingerprint Biometric.

Notably, newborn babies' fingerprints are now captured in hospitals to prevent interchangeability errors, storing the prints of both the mother and child together. This practice is also adopted by old-age pensioners and various public services, such as railways and electricity boards, where fingerprinting of temporary laborers safeguards against muster role frauds. In contemporary settings, fingerprints serve as a means to access computers, with software-equipped locking systems securing bank strong rooms and household doors, preventing unauthorized entry. The widespread adoption of fingerprint recognition systems reflects their high demand across diverse applications.

1.3.2 Types of Fingerprints

Throughout history, three categories of fingerprints have been identified: latent prints, patent prints, and plastic prints. Latent prints refer to intentionally collected fingerprints from individuals, whether for enrollment in a system or during arrest for a suspected criminal offense. Live Scan or ink on paper cards are commonly used methods for collecting latent prints, as illustrated in Fig. 1.4 (a).

Patent prints are accidental friction ridge impressions that are easily visible to the naked eye, resulting from the transfer of foreign material from a finger onto a surface. Examples include impressions from flour and wet clay. Due to their inherent visibility and lack of need for enhancement, patent prints are typically photographed rather than lifted, as is done with latent prints. Materials like ink, dirt, or blood can leave patent prints on surfaces, as shown in Fig 1.4 (b).



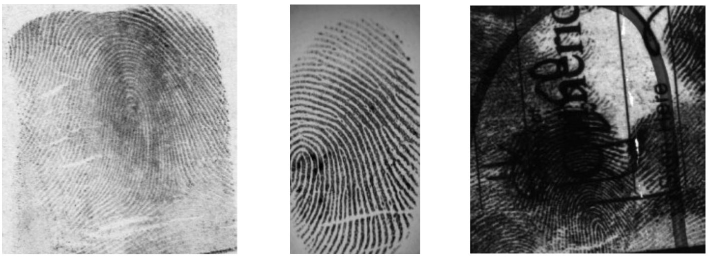
(a) (b) (c)

Figure 1.4: Example of representative fingerprints. (a) Latent ink fingerprint. (b) Patent fingerprint. (c) Plastic fingerprint.

A plastic print is a friction ridge impression embedded in a material that preserves the detailed ridge shape. While it is rare for criminals to leave prints in wet clay, this material serves as a perfect medium for creating plastic prints [32]. Examples commonly encountered include melted candle wax, putty removed from windowpane perimeters, and thick grease deposits on car parts. Unlike latent prints, plastic prints are already visible and require no additional enhancement (see Fig 1.4 (c)).

Regardless of the aforementioned three types, the collected fingerprints can be categorized into Rolled/full, Plain/flat, and Latent or partial (see Fig. 1.5). Rolled fingerprint images are obtained by rolling a finger from one side to the other ("nail-to-nail") to capture all the ridge details. Plain impressions occur when a finger is pressed down on a flat surface without rolling. Rolled and plain impressions can be obtained through the scanning of inked impressions on paper or by using live-scan devices. These attended mode acquisitions typically result in high-quality fingerprints rich in information content.

On the other hand, latent or partial fingerprints are lifted from surfaces inadvertently touched or handled by a person through various methods, ranging from simple photographing of the print to more complex dusting or chemical processing [33], [34]. The critical aspect in forensics is the matching of a latent fingerprint against a database of rolled prints or latent prints (reference prints) to apprehend a criminal.



(a) Rolled (b) Plain (c) Latent

Figure 1.5: Types of Fingerprints.

1.3.3 Partial Fingerprints

Automated fingerprint recognition, primarily focused on complete fingerprints, has gained widespread usage and constitutes a well-explored field of study. However, the realm of partial fingerprint recognition emerges as an intriguing research area, involving the matching of a fragment of a fingerprint with a template or reference of pre-enrolled fingerprints. A partial fingerprint is defined as an incomplete fingerprint, and several examples are depicted in Fig. 1.6.

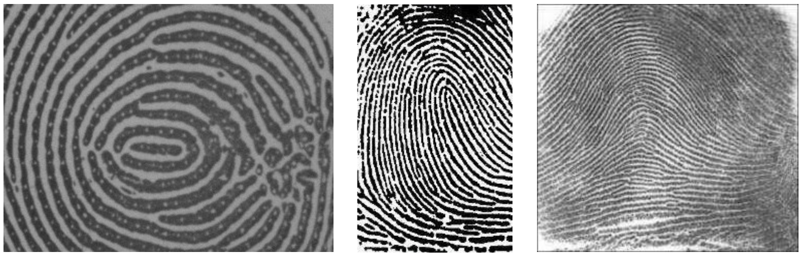


Figure 1.6: Examples of Partial Fingerprints.

Matching partial fingerprints against full pre-enrolled images in a database introduces several challenges [35]:

- The limited number of minutiae points in such prints diminishes their discriminatory capability.
- Uncontrolled impression environments result in unspecified orientations of partial fingerprints.
- Distortions, such as elasticity and humidity, introduced by the characteristics of human skin, can amplify ambiguity between genuine and imposter fingerprints.

The distinct characteristics inherent in partial fingerprints make the implementation of a fingerprint identification system based on partial prints a more formidable challenge compared to conventional AFRS.

1.4 Fingerprint Features

The fingers are covered with a specialized type of skin characterized by minute elevated lines known as 'Papillary ridges' or 'Friction ridges.' The depressions situated between these elevated ridges are referred to as 'Forrows' or 'Valleys.' Additionally, white lines known as 'creases' may also be present. Notably, the friction skin area of a finger remains unobstructed by hair or glands. A fingerprint is essentially a reproduction of the friction skin surface on the first phalanx of the finger. In impressions made with black printer's ink, ridges are represented as 'black lines,' while valleys are depicted as 'white lines.'

The features that can be gathered from a fingerprint are classified into three groups [4], namely:

- Level 1 features: Encompasses macro details such as ridge flow and pattern (see Fig. 1.7 (a)).
- Level 2 features: Involves points or minutiae details (see Fig. 1.7 (b)).
- Level 3 features: Encompasses high-level or micro details like pores and ridge contours (see Fig. 1.7 (c)).

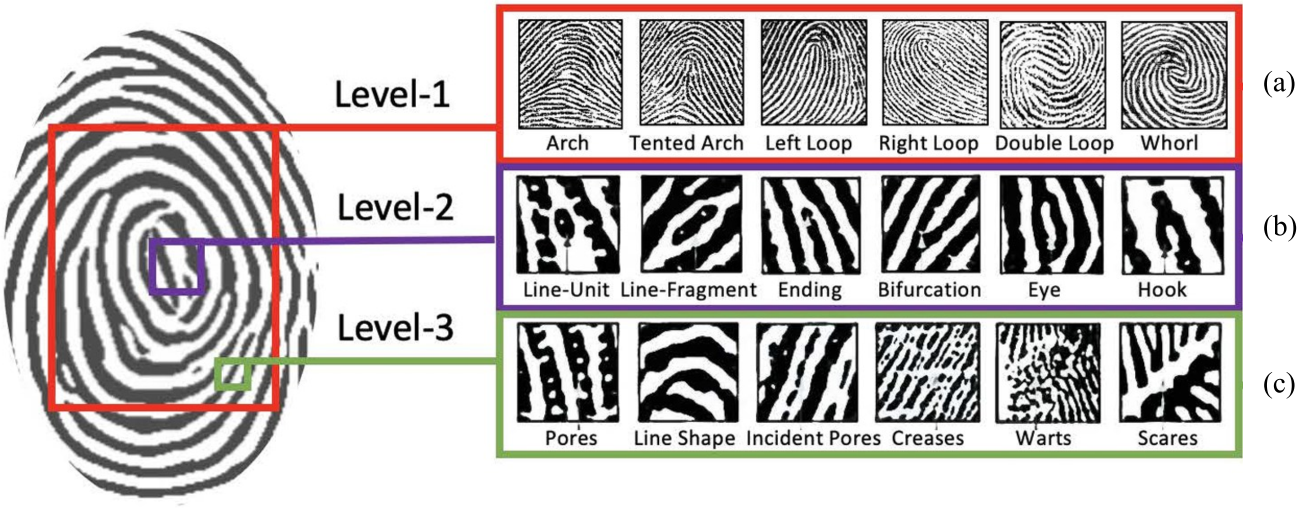


Figure 1.7: Fingerprint Features.

1.4.1 Level 1 Features

Level 1 features encompass macro details, providing information about the general ridge flow and pattern configuration. These features include orientation details, core and delta locations, and the ability to distinguish between finger and palm prints. At a global level, the fingerprint pattern reveals distinctive regions characterized by high curvature and frequent termination. Level 1 features comprise these global patterns and morphological information. While not sufficient for unique fingerprint identification, they serve for broad classification and exclusion. Examples include arch, tented arch, left loop, right loop, double loop, and whorl (see Fig. 1.7 (a)).

1.4.2 Level 2 Features

The comprehensive analysis of fingerprints for matching purposes necessitates the thorough comparison of numerous features embedded in the print pattern [36]. These features encompass distinct patterns intrinsic to the fingerprint, which further manifest in various unique characteristics. Within a fingerprint, several specific features stand out, such as islands (a standalone line that does not connect with other regions), dots (an independent ridge with roughly equal length and width), crossovers or bridges (short ridges running between parallel ones), core (the central point of the fingerprint pattern), and delta (the singular point from which patterns deviate).

Patterns originating from ridges are collectively referred to as 'Minutiae,' with two primary forms known as 'ending' and 'bifurcation.' An ending denotes a feature where a ridge concludes, while bifurcation represents a feature where a ridge diverges from a single path into two paths at a Y-junction. These two types give rise to several other feature classifications, including lake or enclosure features (a single ridge point that bifurcates and reunites shortly after to form a single ridge) and spur (a bifurcation with a short ridge branching off a long ridge). Additionally, features like line-unit, line fragment, eye, and hook can also be extracted. Collectively, these features are termed low-level features (refer to Fig. 1.7 (b)) or level 2 features, and they are easily extractable from a fingerprint image.

Level 2 features pertain to the diverse manners in which ridges exhibit discontinuity. Minutiae, being the most prominent features, possess heightened stability and robustness. The distribution of minutiae features is acknowledged as distinctive to individuals, and empirical evidence substantiates their substantial discriminatory power in establishing the uniqueness of fingerprints. It is imperative to acknowledge that achieving higher accuracy during recognition necessitates extracting a considerable number of minutiae points.

1.4.3 Level 3 Features

Level 3 features encompass fine intra-ridge details, including dimensional attributes of a ridge, such as ridge path deviation, width, shape, pores, edge contour, incipient ridges, breaks, creases, scars, and other permanent details [37] (see Fig. 1.7 (c)). Pores and ridge contours are commonly used level 3 features, where the ridge contours provide valuable information on ridge width and edge shape. The shapes and relative positions of ridge edges are considered permanent and unique. This research will specifically emphasize the pore feature.

1.4.4 Pore Features

Most automatic fingerprint recognition systems primarily utilize Level 1 and Level 2 features, as they are evident in images captured with a resolution of at least 500 dots per inch (DPI). In contrast, Level 3 features can be derived from high-resolution fingerprint images ranging from 800 to 1000 dpi [38], encompassing pores, incipient ridges, and edge contours. The coordinates of pores, known for their high discriminatory value [39], contribute significantly to fingerprint recognition technology employing Level 3 features, showcasing elevated accuracy. Pore information is also instrumental in partial matching [40] and fingerprint liveness detection [41], particularly when combined with Level 2 minutiae features (refer to Fig. 1.8).

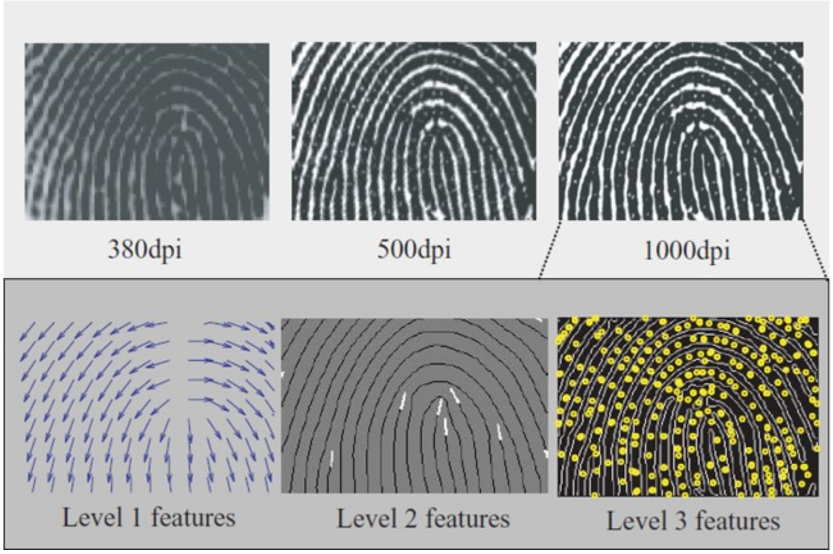
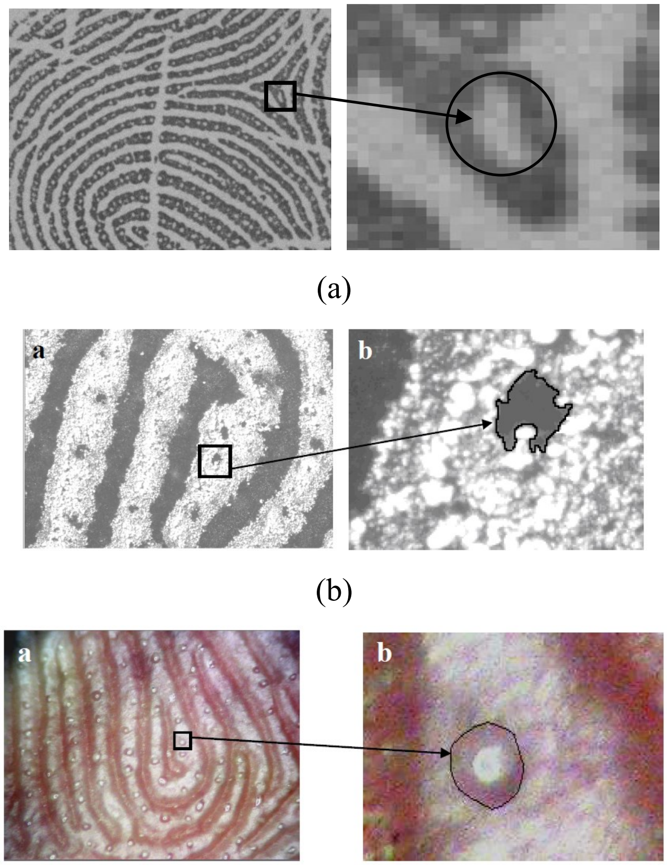


Figure 1.8: Levels 1, 2 and 3 for fingerprint information from different image resolutions. In level 3, the yellow dots are the pores from the high-resolution image.

With advancements in fingerprint-capturing techniques, increased image resolution has become crucial. Higher image resolution facilitates the inclusion of level 3 features in fingerprint recognition, proven to be as unique and permanent as level 2 features. Pores have been statistically analyzed and validated for personal identification, especially beneficial in cases where only partial fingerprints are captured, and minutiae points are insufficient [42], [43].

The complex shapes of pores in touched-based, touchless, and latent fingerprints pose a challenge, as illustrated in Fig. 1.9. The varying shapes of pores among individuals and even within the same person necessitate the use of features such as pore spacing, frequency, and total number for characterization. Pore features gain acceptance, particularly with high-resolution fingerprint scanners enabling reliable pore feature extraction. Pores, a popular level-3 feature, serve as useful supplementary features in forensic applications.



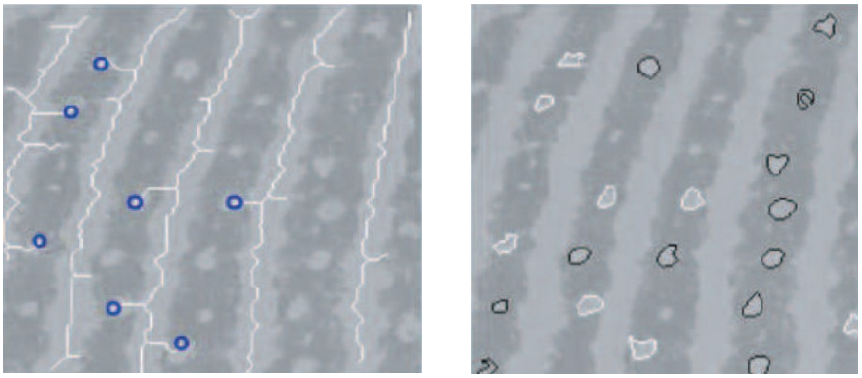
(a)

(b)

(c)

Figure 1.9: Pore on different types of fingerprints. (a) Touched-base fingerprint. (b) Latent fingerprint. (c) Touchless fingerprint.

Pores serving as the outlets for perspiration-secreting glands. These pores contain a combination of oily matter, sweat, and other secretions, contributing to the smoothness, lubrication, and softness of the hand. Arranged uniformly along the summit of the ridge, pores are persistent and unalterable, forming the focal point of interest in this research. Studies indicate a pore density on ridges ranging from 23 to 45 pores per inch, with 20-40 pores deemed sufficient for successful individual identification [39]. Pores exhibit varying sizes, ranging from 88 to 220 microns in diameter. Two categories of pore features are distinguished based on perspiration activity (see Fig. 1.10): (i) Open pores, intersecting with the valley between two ridges, and (ii) Closed pores, entirely enclosed by a ridge.



(a) (b)

Figure 1.10: Pore classes. (a) Open and (b) closed pores.

In forensic applications, pores offer advantages over minutiae [44], being less susceptible to damage or mimicry and abundant even in small fingerprint fragments (see Fig. 1.11). These advantages motivate researchers to incorporate pores into partial fingerprint recognition systems, showcasing their potential in enhancing the robustness and security of fingerprint recognition.

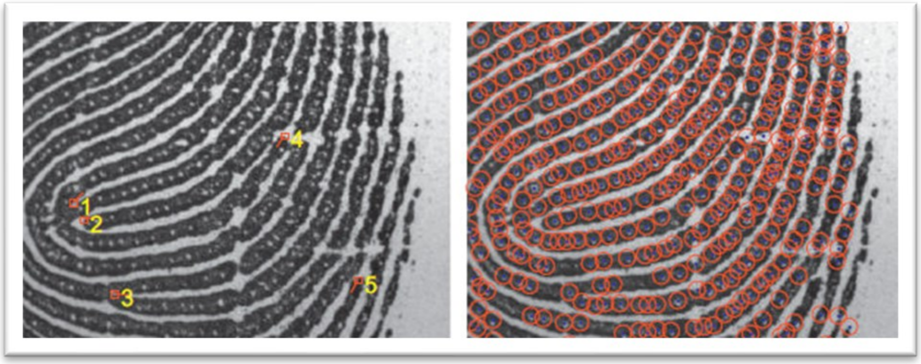


Figure 1.11: Minutiae (left side) and Pores (right side) counts on a partial fingerprint.

1.5 Fingerprint Recognition

Fingerprint recognition has gained widespread popularity due to its ease of acquisition, diverse data sources, and extensive utilization by law enforcement and immigration agencies [3]. It serves as a fundamental aspect of biometric authentication and has undergone significant advancements over time. Researchers have delved into various levels of features to improve accuracy and reliability.

Initially, focus was placed on ridge patterns, which provide a broad structural overview of fingerprints, including ridge orientation. While these patterns were easy to extract from low-resolution images, they lacked detailed information and could lead to false positives, particularly among individuals with similar ridge patterns such as identical twins or family members. Moreover, ridge patterns often failed to capture local variations caused by factors like scars, cuts, or aging, which could significantly impact recognition accuracy, especially in noisy or distorted images.

In response to these limitations, minutiae-based approaches gained prominence. Minutiae, comprising ridge endings and bifurcations, offered a more specific and reliable means of fingerprint recognition. However, they too had drawbacks, particularly in cases where the captured fingerprints lacked sufficient minutiae or exhibited non-standard ridge patterns like loops or whorls.

The combination of ridge patterns and minutiae sought to mitigate their individual shortcomings, providing both a coarse overview and finer details [45]. Nevertheless, this approach requires a balance between robustness and precision, and results in increasing the template size, storage needs, and matching complexity.

The introduction of Level 3 features, specifically pores, represents a continued effort to enhance fingerprint recognition. Pores offer unique attributes for each individual and have shown promise in improving recognition accuracy, especially with the availability of high-resolution scanners capable of capturing detailed fingerprint images.

In recent years, the development of fingerprint recognition schemes using pores has drawn a great deal of researcher's attention in view of the availability of high-resolution scanners that are capable of providing very good quality fingerprint images. A pore-based AFRS is essentially a pattern recognition system that has mainly two modules [4], [38], [46]. In the first module, the pores of the fingerprint are detected using some relevant features of these pores. In the second module, fingerprint pores are utilized for recognition by extracting the relevant features from the detected pores and using these features to match the pores between the query and template fingerprints. In order to have a reliable pore-based AFRS, one needs to have both a good pore detection module that can detect the pores precisely as well as a module that can extract, represent and match the fingerprint pores accurately.

Fingerprint recognition based on pore features has traditionally relied on handcrafted algorithms to detect pores and match them between the query and template fingerprints [38], [47], [48], [49], [50], [51], [52], [53], [54]. However, recent research has focused on the use of neural network algorithms to improve the accuracy and efficiency of fingerprint recognition systems [55], [56]. Neural networks can learn to extract features from fingerprint images that are difficult to identify using traditional methods. The use of deep learning techniques has enabled the development of more robust and scalable fingerprint recognition systems that can handle large datasets and complex features.

Neural networks, particularly Convolutional Neural Networks (CNNs) [57], [58], [59], have demonstrated their proficiency in feature learning and pattern recognition tasks, making them well-suited for analyzing intricate details such as pore structures [60], [61], [62], [63], [64]. Despite the promising potential of incorporating pore features and neural networks in fingerprint recognition, there are challenges to overcome. Fine-tuning neural network architectures and ensuring robustness to variations in image quality remain focal points for ongoing research. The integration of neural networks into pore-based fingerprint recognition systems opens doors to real-time diverse applications, including forensic analysis, access control, and mobile device security. In the next section, we will review the related works for pore-based Fingerprint recognition system for both traditional and non-traditional schemes.

1.6 Literature Review of Pore-Based Fingerprint Recognition

An automatic fingerprint recognition system is essentially a pattern recognition system that has mainly two modules [4], [38], [46], as shown in Fig. 1.12. In the first module, pores of the fingerprint are detected using some relevant features of the pores. In the second module, the detected pores of the fingerprint are then used to do the actual fingerprint recognition.

The process of pore detection used in various schemes can, in general, be divided into two parts. In the first part, a pore intensity map is obtained by extracting relevant pore features from a gray-level input fingerprint image. A pore intensity map is essentially a collection of gray-level blobs characterizing the pores of a fingerprint image. In the second part, first, the locations of the candidate pores are found from the pore intensity map by locating the coordinates of their centers, and then, by using some prior knowledge about the true fingerprint pores, spurious pores are removed from the set of candidate pores in order to obtain the final set of true pores.

The pore feature extraction and matching module has three segments: in the first segment, relevant features of the detected pores suitable for fingerprint recognition are extracted. In the second segment, the extracted features from the first segment are used to match the pores in the query fingerprint image with a template fingerprint image in the database. In the third segment, the matched pores are refined to remove any false matched pair of pores.

In the following subsections, we will delve into the various methodologies employed for each part of the pore detection module and each segment of the pore feature extraction and matching module, while also providing a critical analysis of these methods. We will examine the approaches utilized in pore detection, including the extraction of relevant pore features and the elimination of spurious pores, aiming to achieve precise detection of true pores within fingerprint images. Additionally, we will explore the techniques employed in the pore feature extraction and matching module, covering the extraction of suitable features from detected pores, the matching process between a query and template fingerprint images, and the refinement of matched pores to eliminate false pairs. Through this comprehensive discussion and critique, we aim to gain insights into the strengths and limitations of existing methodologies in the realm of pore-based automatic fingerprint recognition.

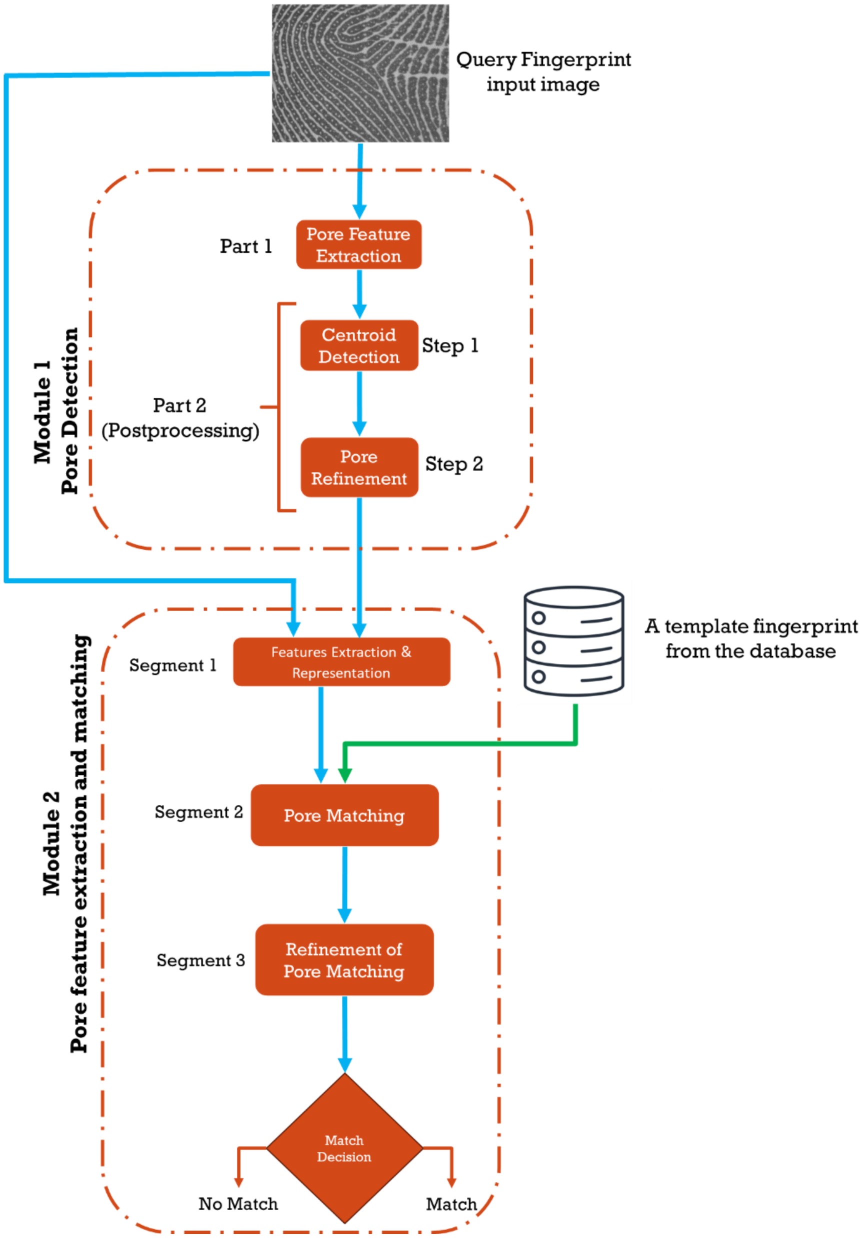


Figure 1.12: General pore-based fingerprint recognition system.

1.6.1 Pore Detection Module

Recent studies proved that Level 3 features can greatly increase the accuracy of current fingerprint recognition technologies [65], [66]. In particular, the number and coordinates of the pores were demonstrated to be highly discriminative features [38], as well as in the case of partial fingerprints [67]. The process of pore detection, in general, has two parts, as shown in Fig. 1.12. In the first part, the possible candidate pores are determined from the gray-level image of the fingerprint by extracting the pore features. In the second part, first, the location of the candidate pores is found by locating the coordinate of their centers and then a decision is made as to the true pores by identifying and removing any possible spurious pores from the set of the candidate pores. Next, we will discuss the background work of each part of the pore detection module.

A. Feature Extraction

The most critical step in the first part of pore detection is the feature extraction step. Traditionally, in the first part, the pore features generated and extracted are handcrafted features. The works in [47] and [48] are the two earliest methods of generating handcrafted pore features, in which the ridges of a binary skeletonized version of the fingerprint image are tracked and any discontinuity along the ridges is considered to represent the presence of a pore when certain criteria are satisfied. In the first part of other traditional schemes for pore detection, the pore features are first modeled in terms of the orientations of the ridges containing the pores and the pore shapes and sizes, and then a filter, such as Gabor [38], adaptive DOG or DAPM [49], is designed to extract the true pores from the fingerprint images. The pore detection performance of the schemes in which the pore feature extraction in the first part is based on the handcrafted features is very much limited by the modeling capability of the pore features that represent only certain types of pores accurately.

The pore detection scheme of [60] is the first CNN-based scheme, in which the first part is based on a simple CNN architecture. In this architecture, two convolutional layers are employed using, respectively, 30 filters and one filter, each of kernel size 5×5, and a max-pooling layer with a 3×3 kernel in between the two layers. However, this scheme uses only a shallow network, which relies only on its width to extract the pore features. Hence, it is not able to learn the deep high-level features of the pores.

The scheme of [61] is another CNN-based pore detection scheme in which a deep network, referred to as Deplore, consisting of 10 convolution layers is used. Each of the layers in this network with the exception of the last one (the reconstruction layer) uses 64 filters of kernel size 3×3. In view of the fact that this scheme uses a deep network, it is capable of extracting high-level pore features and providing performance superior to that provided by the scheme of [60]. However, this scheme is not able to improve the performance further by making the network deeper, since the network cannot handle the gradient vanishing problem associated with a very deep network.

The pore detection scheme of [62] uses a very deep CNN, referred to as DeepResPore, in its first part. This network employs a total of 18 convolutional layers, with the first one using 64 filters each of kernel size 7×7 and the last one (the reconstruction layer) employing a single filter with kernel size 3×3. The main body of this network consists of 8 residual blocks each containing two convolutional layers. The kernel size of each of the filters used in the residual blocks is also 3×3. The number of filters employed in each of the convolutional layers of the first two residual blocks is 64 and these numbers are doubled in the convolutional layers of the succeeding pairs of the residual blocks. The network of this scheme is able to handle the gradient vanishing problem in view of its residual architecture. Thus, by using a larger number of filters for each layer, this network is able to provide a performance much superior to that provided by DeepPore. However, this improvement in performance is achieved at the expense of very large complexity in terms of the number of parameters and multiply-add operations.

B. Pore Centroid Detection and Refinement

As mentioned earlier, in the second part of a pore detection scheme, the coordinates of the centers of the true pores are determined from the pore intensity map obtained in the first part. It is in this part of the pore detection scheme where our knowledge of some important characteristics of pores and pore fingerprint images can be utilized more efficiently without using a CNN network in order to obtain accurate values for the centroids of true fingerprint pores. In the pore detection schemes of [38], [47], [48], [49], [60], the pore intensity map is first binarized using a global threshold value, which converts the pore intensity map into a map of white and black regions. Then, the center of each white region in the binary map is computed and it is considered to be the coordinates of the center of a fingerprint pore. In the pore detection schemes of [61] and [62], the pore intensity map is first partitioned into windows. Then, a pixel with a maximum intensity inside a window is considered to be the center of a fingerprint pore if its intensity is found to be larger than a threshold value. In the second part of the schemes of [38] and [49], the pore detection process is further refined by identifying and removing the false pores. A major disadvantage of the second part of all the existing pore detection schemes, traditional or non-traditional, is that all the available knowledge regarding the fingerprint pores are not taken into consideration and even the knowledge that is taken into consideration is not used optimally.

1.6.2 Pore Feature Extraction and Matching Module

As mentioned earlier, the second module has three segments: feature extraction and representation, pore matching and pore matching refinement. The focus of this module is how to design and implement a pore feature extraction and matching module that leads to a low-complexity, high-accuracy fingerprint recognition system. It is with this overall objective and with reference to Fig. 1.12, that we now briefly review the existing literature on the various components of Module 2.

A. Feature Extraction and Representation

The existing methods in the literature for the pore feature extraction and representation segment (Segment 1) of Module 2 can be classified into two categories depending on whether the pore features extracted by this segment are handcrafted or automatic. In the first category, methods, such as correlation-based method [50], TD-spare method (TDSWR) [51], local descriptors and spatial relations based method [52], RootSIFT feature and edge descriptor method [53] and spatial pairwise co-occurrence descriptor method [54], have been proposed for pore feature extraction and representation. However, these handcrafted pore feature extraction methods, in view of skin distortion or imperfect acquisition of the fingerprint images, do not provide satisfactory pore representations in the sense that the same pore from two different images of the same finger captured at different times may not have the same representation or the pores from the images of two different individuals may have the same representation.

The methods in which pore representation is based on automatic feature extraction of the fingerprint using CNN constitute the second category of the pore feature extraction and representation methods. Recently, in [63], the authors have designed a CNN architecture, referred to as PoreNet, using residual blocks to automatically extract pore features and represent them suitably for fingerprint recognition. Their model using 4 residual blocks generates a 1681-dimensional feature representation of the fingerprint pores. However, the performance of the network is not high compared to some of the handcrafted feature methods, since the use of only 4 residual blocks does not result in producing deep-level features. Increasing the number of residual blocks to produce deep-level features, however, is not a practical solution for this network. This is in view of the fact that this network is constrained to produce the feature maps in different residual blocks to have the same fixed spatial size, and therefore, increasing the number of residual blocks would result in increasing the number of floating-point operations (FLOPs) exponentially.

The authors in [64] have designed a CNN architecture, known as DeepPoreID, to generate 128-dimensional feature representations using 8 residual blocks. This CNN architecture, in view of its deep-level feature extraction capability, has been shown to provide a fingerprint recognition performance that is much superior to those provided by the handcrafted feature-based techniques. At the same time, the complexity, in terms of the number of operations, of this network is only about one-fourth of that of the architecture of [63]. This reduction in complexity is achieved by introducing a max-pooling layer after each residual block, which reduces the spatial size of the channels input to the succeeding residual block. Although the DeepPoreID network has a substantially reduced number of operations compared to that required by the PoreNet network, it still does not outperform the latter for the test images of all the fingerprint datasets consistently.

For mobile applications, such as online banking, a low-power biometric system is a requirement. Also, for forensic applications, a real-time biometric is an essential requirement. In view of these requirements of biometric systems, it is essential to design an automatic feature extraction network that is capable of extracting highly representable pore features and consuming a substantially lower number of arithmetic operations in comparison to that required by the existing feature extraction networks.

B. Pore Matching

In the pore matching segment (Segment 2), the pores of a pair query and template fingerprint images are matched based on the similarities of the pore feature representations obtained from Segment 1. The existing pore matching algorithms can be divided into two categories: alignment-based pore (AP) matching algorithms, such as correlation-based algorithm [68] and iterative closest point (ICP) algorithm [38], and direct pore (DP) matching algorithms, such as TD-Sparse algorithm [51], RootSIFT algorithm [53], LTPM algorithm [54] and the algorithms of [63], [64].

In an alignment-based pore matching algorithm, given a pair of query and template fingerprint images, first, two parameters, namely, the translation and the rotation parameters, are estimated, which when applied to the query image translates and rotates it with respect to a reference point, such as a singular point or a minutia point, so as to provide a maximum overlap between the two fingerprint images. Then, each pair of the pores from the two fingerprint images in the overlapped region or the segments of the overlapped region in a neighborhood of the landmarks are matched using a geometric distance, such as Euclidean distance, between their feature representations. However, the matching of the fingerprint images in AP matching algorithms is dependent on the accuracy of the detection of the landmarks and the alignment of the fingerprint images, which is affected in view of the presence of noise and distortions in the query and template fingerprint images. Moreover, the alignment segment of the schemes in this category has an overhead of consuming more time. On the other hand, in the pore matching schemes belonging to the DP matching category, all the pore pairs from the two fingerprint images are matched without the alignment of the two images, resulting in an improved accuracy of pore matching. It is to be noted that the increase in time resulting from the comparison of the pores in a larger number of pairs is only a small proportion of the time taken by the process of aligning the fingerprint images in the schemes of the first category.

In order to determine the similarity between the pores of a pair from the query and template fingerprint images, a metric for the similarity between the pore feature representations of the two pores is used. For instance, in all the pore matching algorithms as mentioned above in the two categories, a metric of geometric distance has been used. However, the geometric distances between the pore feature representation of a given pore in the query image and the pore feature representations of more than one pore in the template image may be the same or very similar. Hence, the use of a single metric (for example, a metric of distance) to determine the similarity between the feature representations in the query and template fingerprint images is not adequate. Consequently, the use of more than one metric on the pore feature representations should improve the matching accuracy between a pair of pores.

C. Pore Matching Refinement

The features of a pore in the query fingerprint image may be very similar to that of one or more pores in the template fingerprint image. As a result, a pair of pores resulting from Segment

2 of the pore feature extraction and matching module may lead to a situation in which a pore in the query image is paired with one or more non-corresponding pores in the template image. This is in view of the fact that different pore patches from a different region of two images of different fingers can look very similar, as the example (red circled) shown in Fig. 1.13. Hence, most of the pore feature extraction and matching modules have a pore matching refinement segment (Segment 3) whose purpose is to eliminate as many as possible all such pairs from the pore correspondence set obtained in Segment 2.

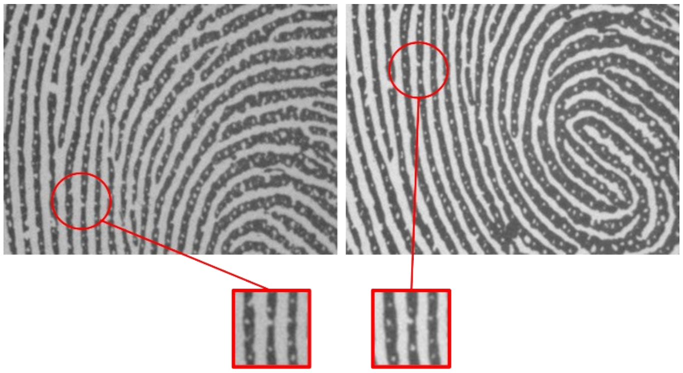


Figure 1.13: An example of imposter fingerprint pairs with similar pore patches.

The existing pore matching refinement schemes can be categorized into two classes based on the way these schemes attempt to eliminate the pairs of inaccurately matched pores. The schemes [50], [51], [64] of the first category are position-based. In this category, the position (coordinates) of a very small subset of pairs from the matched pores from the pore correspondence list is randomly selected and used to build a linear model. This linear model is then applied to all the other pairs, one by one, from the pore correspondence list. Then, all the test pairs (i.e., the remaining pairs from the corresponding list) that fit this model and also satisfy certain threshold conditions are considered to be inliers with respect to this model. The process of randomly choosing a small subset of pairs, building a linear model and testing it to determine the corresponding inliers is repeated a prespecified number of times. Finally, the model that provides the maximum number of inliers is chosen to refine the pore correspondence list. The disadvantage of the refinement schemes in this category is that the linear models built do not provide the refined list of correspondence in which all the pairs are necessarily true matches. As a matter of fact, it has been shown that the refinement schemes in this category do not perform well, from the accuracy point of view, if the number of outliers in the pore corresponding list resulting from the pore feature extraction and matching module of Segment 2 exceeds 50%. Moreover, building and testing a large number of models in these schemes is time-consuming.

The second category of pore matching refinement schemes can be referred to as graph-based in which two graphs, one corresponding to the pores in the query image and the other corresponding to the pores in the template image, are formed and used to eliminate the pairs of the pores that are falsely matched in the correspondence list obtained from Segment 2. In the schemes of [52], and [53] of this category, one graph is formed using the positions of the pores in the query image as nodes of the graph, and in this graph, an edge between a pair of nodes is introduced only if the Euclidean distance between the nodes falls within a prespecified range. The other graph is formed corresponding to the pores in the template image. In this second graph, as in the query image graph, the positions of the pores form the nodes of the graph and an edge between a pair of its nodes is introduced if and only if an edge exists between the corresponding pair of nodes in the graph of the query image. In the schemes of this category of pore matching refinement, a metric to measure the similarity between the two corresponding pairs of edges of the two graphs is defined and used to refine the correspondence list.

In comparison with the schemes in the position-based category, these two graph-based pore matching refinement schemes are both more accurate and less time-consuming. However, the applicability of these schemes is limited to those pore-based fingerprint recognition systems in which its pore matching segment (Segment 2) leads to a one-to-one pore correspondence list. In order to improve the speed and accuracy of the graph-based techniques of [12] and [13], another graph-based scheme has been proposed in [54] to refine the pore correspondence list. In this scheme, for a given pair of pores (P_{Qi}, P_{Ti}) from the pore correspondence list, local graphs are formed individually for the pores in the query and template images. The local graph for the query image consists of a node for the pore P_{Qi} and those that are in the K-nearest neighborhood of P_{Qi}, but excluding those pores from this neighborhood that also have the same matched pore in the template image as that of P_{Qi}, i.e., P_{Ti}. The local graph corresponding to P_{Qi} for the query image is formed by placing an edge between P_{Qi} and other nodes in the neighborhood pores. The local graph for the pore P_{Ti} in the template image is formed by selecting all pores corresponding to the ones used in the local graph of P_{Qi} and placing edges between the pairs in the same way as done for performing the local graph for the query image. Therefore, in this scheme, we have as many pairs of local graphs as the number of matched pairs of pores. In the calculation of the similarity measure between the pair of pores P_{Qi} and P_{Ti}, the angles of the edges and Euclidean distances between the pore i and each of its neighborhood pores are used and the same goes for the pore j. In this scheme, the structure of the local graphs allows the use of spatial neighborhood information in the calculation of the similarity measure in a rotationally invariant manner. This graph-based scheme for pore matching refinement is capable of removing the mismatches more accurately and efficiently in comparison to the other two graph-based schemes.

1.7 Problem Statement and Motivation

Fingerprint recognition technology has various advantages such as offering high security, less cost and a non-invasive manner of acquisition and therefore is one of the most frequently used mechanisms. Owing to the high individualism and permanent nature portrayed by fingerprint, this biometric is chosen in the present research work. Advanced information technology and demand for high security have forced academicians and researchers alike to search for different recognition methods that are accurate, economical, fast, and safe. Fingerprint recognition is emerging as one of the important areas in biometrics authentication. Matching fingerprints continues to be an important challenge today, despite the advances made in fingerprint recognition techniques. The development of partial fingerprints is further challenged by the uncontrollable impression environments, resulting in the introduction of unspecified distortions like skin cuts or acquisition device quality.

The main challenge in designing an effective fingerprint recognition system is the task of satisfying three extremities (i) the simplicity and ease to use. (ii) the scalability of working with small and high-size images. (iii) the acceptability in producing maximum accuracy. This combination is very challenging to achieve in fingerprint recognition systems. The most frequently used feature in fingerprint recognition algorithms is the minutiae features. A matching algorithm returns the number of matched minutiae on both query and template fingerprints, which will then be used to generate similarity scores. The similarity score increases when the number of minutiae features matched is high. That is, when the number of minutiae on both fingerprints is large, the ability of the system to confidently distinguish the genuine and imposter fingerprint is also high. According to forensic guidelines, when two fingerprints have a minimum of 12 matched minutiae they are considered to have come from the same finger [4], [69].

However, while considering partial fingerprints, the number of minutiae detected, and the size of the fingerprint are directly proportional. That is, when the fingerprint size is small, the feature detector can detect only a smaller number of minutiae. This would lower the similarity of the minutia matching score and consequently reduce the recognition accuracy. For example, consider the partial fingerprint in Fig. 1.14. There are only 5 points of detected minutiae and the ridge densities of the partial and full prints are equivalent in the area of overlay, which is not enough for reliable authentication.

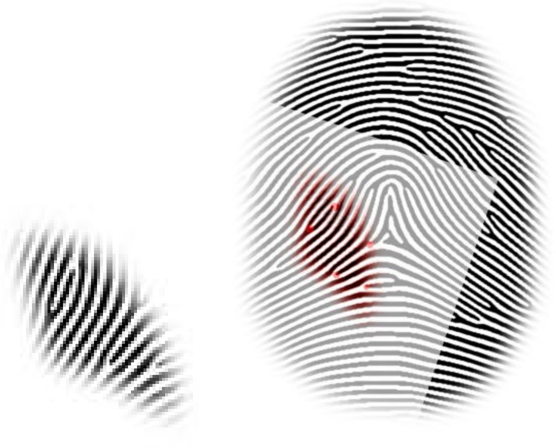


Figure 1.14: Partial and Full Fingerprint.

This clearly shows that the relation between the acquired fingerprint size and the security strength plays a key role in designing a fingerprint recognition system and hence needs to be studied carefully. In particular, the demand for high security has increased the search for other efficient features that can be used to increase accuracy while decreasing the complexity of partial fingerprint recognition systems.

Exploring more features in addition to or apart from minutiae on fingerprints is a recent trend and the benefits of using pores in fingerprint recognition have also been studied and validated [38], [44], [47], [47], [50], [51], [52], [53], [54], [60], [61], [62], [64], [68], [70], [71], [72], [73], [74]. The main modules in pore-based fingerprint recognition are pore detection and pore feature extraction and matching. As the most critical step in both stages is the feature extraction step, most research was focused on using traditional or handcrafted features. However, the performance of such schemes is very much limited by the modeling capability of the pore features. Recently, machine learning algorithms are considered the most appropriate for classifying an image. A neural network is a well-known model designed to work on human neurons and is more refined, accurate and reliable in accomplishing problems of classification. Neural networks attempt to imitate the way the human brain works, which makes them the right candidate for use with fingerprint recognition. However, the improvement in the performance of recent research for fingerprint pore-based recognition using machine learning is achieved at the expense of very large complexity and time consumption.

Personal identification or recognition is required in applications such as online banking using mobile devices. Thus, a low-power architecture is necessarily a requirement for the deployment of a fingerprint recognition scheme. In biometric recognition applications, such as forensic applications, the algorithm must be run in real-time. These requirements of the biometric applications need the development of biometric identification schemes, including those based on the pores of fingerprint images, with very low complexity and real-time consumption.

It is in a hybrid (non-traditional) pore detection module where one can more efficiently exploit the strength of CNNs in extracting automatic pore features and also use the knowledge base on the characteristics of the pores and fingerprint images. CNN-based networks for pore feature extraction and matching module struggle to produce deep-level features effectively due to limitations in residual block utilization and complexity considerations. Moreover, existing pore matching methods primarily rely on Euclidean distance metrics, highlighting the need for alternative criteria exploration. Enhancing these methodologies could substantially improve the accuracy and reliability of pore-based fingerprint recognition system.

1.8 Objectives

Based on the motivation provided in Section 1.7, this research aims at developing a low-complexity, high-accuracy pore-based fingerprint recognition system leveraging convolutional neural network. The following objectives are formulated to facilitate the development of two modules, namely, pore detection, and pore feature extraction and matching.

- In the pore detection module, it is in a hybrid (neural network) pore detection module where one can more efficiently exploit the strength of convolutional neural networks (CNNs) in extracting automatic pore features and also use the knowledge base on the characteristics of the pores and fingerprint images. The objective of this stage is to develop a two-part pore detection module in which for the first part a low complexity deep neural network is designed to produce pore intensity maps by focusing on the connectivity of its various hierarchical parts that result in the extraction of meaningful features leading to a highly representational pore intensity map, and for the second part, a method is developed for the pore detection by making efficient use of the knowledge base on fingerprint pores. Since different kinds of features are very important in detecting the pores, the strategy in designing the network is to generate hierarchically low, middle and high-level features and to concatenate them to produce a very rich set of pore features. The features at each of the three levels are learnt locally in a residual framework, which helps in curtailing the gradient vanishing problem [75], [76], [77]. By placing emphasis on the architecture design of the network for generating an enriched set of features, it is possible to keep the number of filters and layers low, which results in a low-complexity network.

- The second part of the pore detection module aims at eliminating the false pores and determining the coordinates of all the true pores accurately from the pore intensity map obtained in the first part in conjunction with the fingerprint image itself. In this part of the pore detection module, the knowledge on fingerprint images, such as the pores intensity, their variation from one region to another region of the fingerprint image and the minimum distance between two neighboring pores, is used to accurately detect the pores of a fingerprint image.
- The accuracy of a neural network-based pore detection module is highly dependent on its ability to extract a very rich set of pore features and augmenting this ability with the known attributes of fingerprint pores in determining their true locations in the image. In this regard, the main contributions of our proposed pore detection module are as follows. (i) A new CNN architecture is developed that combines the features extracted at hierarchically low, middle and high levels. (ii) By recognizing the first part of the proposed pore detection module to be essentially a pixel classification problem, the design of the CNN architecture is carried out using depth-wise convolution that leads simultaneously to a superior binary pore classification [28] as well as to an ultralightweight network. (iii) Since the intensity of a pore pixel relative to that of a non-pore pixel varies from region to region of a fingerprint image, a locally adaptive threshold is used to binarize the pore intensity map. (iv) The center of a pore is obtained as the mean of the intensity-weighted coordinates of the pixels belonging to the pore region in order to obtain more accurate coordinates for the pore centroids. (v) In order to distinguish between a true pore and a false pore (a) a locally adaptive intensity threshold that

is determined based on the local mean and the variance of the intensities of the pixels belonging to the local pore regions is used, and (b) a minimum distance criterion is used, since the fact that a pair of true pores in a fingerprint image satisfy the requirement of certain minimum distance between them.

- In pore feature extraction and matching module, the objective is focusing on the design of the first two segments of the module, i.e., pore feature extraction and representation, and pore matching, to provide good performance with low complexity for the task of fingerprint recognition. In the first segment, a low-complexity residual block is first developed, and then, used in designing a deep convolutional neural network so as to produce highly representable pore features for the point of view of matching the pores of the query fingerprint image and a pore in the template fingerprint image. In the design of the residual block, separable convolutions, instead of standard convolutions, in conjunction with Max-pooling are used, which results in reducing the number of parameters but more significantly the number of arithmetic operations.
- The main ideas in designing the proposed network for pore feature extraction and representation are as follows: (i) Since the proposed residual block is of low complexity, the network can be made deeper by using a larger number of our residual blocks, and yet, the network complexity is kept low. (ii) The final set of pore features are generated by concatenating the features extracted at various hierarchical levels, and hence, they are highly representable. (iii) The network's training is enhanced by using the idea of deep supervision.
- In the second segment of the pore feature extraction and matching module (the pore matching segment), a new metric is proposed to measure the similarity between a pore in the query image and the pores in a template image. The proposed metric is formed by utilizing information such as the angles and magnitudes difference of a pair of vectors representing the features of a query pore and a template pore, in addition to their Euclidean distance. The matching of the pores of the query and template fingerprint images is carried out by minimizing this proposed metric to generate a pore correspondence list.
- In the third segment of the proposed pore feature extraction and matching module, the graph-based pore matching refinement scheme [54] is used to carry out the refinement of the pore matching of the pairs of pores correspondence list, that is, to remove those matches that are falsely matched.

1.9 Organization of the Thesis

The underlying objective of this research work is to develop an efficient CNN-based pore fingerprint recognition system that can meet the security demands of the authentication field. The thesis is organized as follows:

This chapter (Introduction) provided a concise overview of biometric authentication, focusing particularly on fingerprint recognition. It introduces various levels of fingerprint features and underscores the significance of pores for fingerprint recognition systems. Furthermore, the chapter addresses the existing research of pore-based fingerprint recognition that is significant to this research. A critical look at the available works of literature related to the present research work in both pore detection and pore feature extraction and matching is presented in this chapter. Lastly, the motivation and the objective of the research were also debated in this chapter.

Chapter 2 provides an overview of essential background materials pertinent to fingerprint recognition. It covers neural networks, performance metrics, such as accuracy and efficiency, Euclidean distance, and various pore matching refinement algorithms. These topics are used in developing fingerprint recognition systems.

Chapter 3 proposes a two-part scheme for the pore detection module and provides the motivation and rationale behind ideas used in the two parts. This chapter also presents the experimental results of the proposed and other pore detection schemes when applying it to a bench-mark high-resolution fingerprint database. In this chapter, the contribution and the impact of each part of the proposed scheme on the overall performance are investigated and the results are compared with those of the existing pore detection modules.

Chapter 4 introduces a novel pore feature extraction and matching module, which first describes the architecture of the proposed deep convolutional neural network using residual blocks for an automatic extraction of the features relevant to pore matching. Then, the details of designing a novel residual block to be used in the proposed network are described. This chapter also describes the rational and details of a new pore matching metric and use it for matching the pores by using the pore features representation extracted in the previous segment of the proposed module. In this chapter, a number of experiments involving the proposed module for fingerprint recognition are performed. Different ablation studies are carried out to show the effectiveness of the proposed network and that of the proposed metric for pore matching. The performance and complexity of the proposed pore feature extraction and matching module are also compared with those of other existing state-of-the-art schemes.

Chapter 5 concludes the thesis by summarizing the salient features of the proposed CNN-based pore fingerprint recognition scheme. It also explains the possible work to be carried out in the future.

Chapter 2

Background Material

2.1 Introduction

This chapter introduces the concepts of some of the algorithms and methods used in developing fingerprint recognition systems. The chapter starts with the convolutional neural network models inspired by the human brain in Section 2.2; it covers key components like convolution layers, activation functions, and prominent networks such as ResNet and DenseNet. Section 2.3 discusses the evaluation of the system effectiveness, focusing on metrics crucial for gauging system reliability in diverse applications. Section 2.4 describes the significance of Euclidean distance in feature comparison within biometric systems, particularly in fingerprint recognition. Lastly, in Section 2.5, the pore matching refinements schemes, RANSAC, WRANSAC, and LTPM, is discussed and their role in enhancing accuracy and robustness across various fingerprint recognition systems are explained.

2.2 Basic Concepts of Convolutional Neural Network

The human brain stands as the pinnacle of complexity in the realm of computing devices. Its remarkable abilities in thinking, remembering, learning, and problem-solving have inspired studies and modeling efforts in digital computing. Yet, the brain's intricacies remain largely uncharted territory, with scientists merely scratching the surface to understand its workings. Despite the limited knowledge about the brain and neural systems, fundamental attempts to emulate its functionality have given rise to the creation of neural networks.

Artificial neural networks (ANNs) employ similar principles to mimic and harness some of the potency exhibited by the brain and its neural network. Most of the ANNs [64], [65] (e.g., feedforward neural networks) consist of layers of neurons that process data in the form of one-dimensional signals. Examples of one-dimensional signals include feature vectors, time series, measurements, and spatial coordinates. Supervised learning procedures allow neural networks to learn from examples and adapt their inner structure to acquire the capability of generalization, with which the neural network is able to approximate the function also in the case of an unknown input signal [78], [79]. In the majority of ANNs, a feature extraction step computes the one-dimensional input signals [80] from data with higher dimensionality (e.g., an image). The feature extraction step requires a priori knowledge of the problem to efficiently reduce the dimensionality of the input data while maintaining the most significant information.

Convolution Neural Network (CNN) [57], [79], [81] is a type of Artificial Neural Network with several layers. It is an efficient method developed in recent years and has attracted wide attention in the field of Artificial Intelligence. Taking LeNet-5 [81] as an example, as shown in Fig. 2.1, it is a hierarchical neural network with convolutional layers alternated with pooling layers, followed by some fully connected layers.

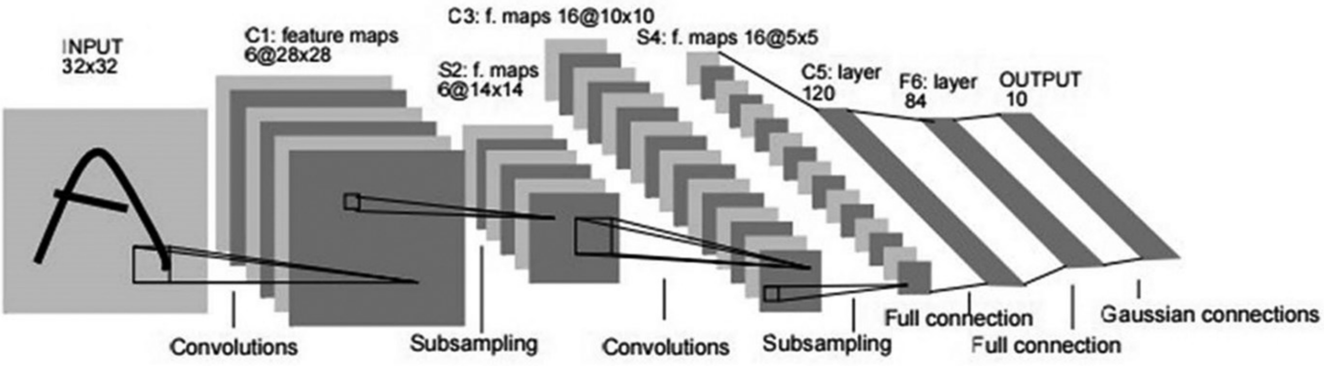


Figure 2.1: Architecture of the LeNet-5 network.

CNNs are a particular form of artificial neural networks whose layers have a structure that permits to process data in the form of multi-dimensional arrays, such as images [79]. In particular, the foremost layers of a CNN are multi-dimensional filter banks that process the input image by convoluting it with the corresponding filter. The convolutional layers can use several types of filters to extract the most significant visual features. The convolutional layer is a feature extraction layer. It utilizes various kernels to convolve the whole image or the former feature map to get new features. The convolutional layer function is donated as $f(.)$ and the feature map can be computed as follows.

$$x_j^l = f\left(\sum_{i\in R_j} x_i^{l-1} * W_{ij}^l + b_j^l\right) \tag{2.1}$$

Here x_j^l is the output at i-th feature map of the l-th layer, (*) is the 2D discrete convolution operator, W_{ij}^l is a trainable filter (kernel) and b_j^l is a trainable bias parameter. Usually, the convolutional layers are followed by an activation function, which plays a crucial role in artificial neural networks by introducing non-linearity into the network's computations. This layer allows it to learn complex patterns and relationships in the data. Without non-linear activation functions, the neural network would reduce to a linear model, severely limiting its capabilities. The choice of activation function depends on the specific task, network architecture, and considerations such as vanishing gradients, computational efficiency, and model interpretability. The activation function impacts the network's learning dynamics and performance, making it a crucial aspect of neural network design and optimization [82].

Generally, a pooling layer follows a convolutional layer to reduce the feature dimension by down sampling the features extracted by convolutional layers. The pooling operation not only reduces the complexity of the convolutional layers but also restrains the phenomenon of over-fitting. Meanwhile, it enhances the tolerance of features to minor distortions, rotations and increases the performance and robustness of deep CNN. Denoting the pooling function as $down(.)$, the feature map after pooling can be calculated as follows.

$$x_j^l = f\left(down(x_i^{l-1}) * w_j^l + b_j^l\right) \tag{2.2}$$

The subsequent layers convert and aggregate these features into a more abstract representation [57]. Among the advantages of CNNs with respect to neural networks able to process only one-dimensional input signals, there is the fact that CNNs require less prior knowledge of the problem. In fact, CNNs do not require a preliminary feature extraction step because a generic convolutional layer can extract the salient visual features from images depicting a great range of objects and situations. For this reason, researchers use CNNs in different application scenarios, such as object classification [57] or natural language processing [81]. In biometrics, there are several recognition schemes based on CNNs for different traits, such as face [83], iris [84], [85], and fingerprint [86]. There are also liveness detection methods based on CNNs and designed for a wide set of biometric characteristics [87], [88], [89].

CNNs are driving major advances in many computer vision tasks, such as image classification [90], object detection [91], [92] and semantic image segmentation [58]. The last few years have witnessed outstanding improvements in CNN-based models. Very deep architectures [75], [90], [93] have shown impressive results on standard benchmarks such as ImageNet [94] or Microsoft common objects in context (MSCOCO) [95] databases. Among CNN architectures extended as fully convolutional networks (FCNs), residual networks

(ResNets) [75] make an interesting case. ResNets facilitate training deep networks by introducing residual blocks, which combine a non-linear transformation with an identity mapping via shortcut connections, enhancing convergence and performance. ResNets are designed to ease the training of very deep networks (of hundreds of layers) by introducing a residual block that sums two signals: a non-linear transformation of the input and its identity mapping. As shown in Fig. 2.2, the identity mapping is implemented by means of a shortcut connection, and their outputs are added to the outputs of the stacked layers which perform the formulation of $F(x) + x$ where $F(x)$ is the mapping after the second layer, x is the input.

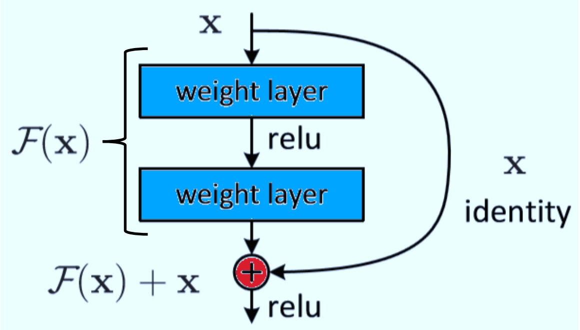


Figure 2.2: ResNet residual block.

ResNets have been extended to work as FCNs [96], [97] yielding very good results in different segmentation benchmarks. ResNets incorporate additional paths to FCN (shortcut paths) and, thus, increase the number of connections within a segmentation network. These additional shortcut paths have been shown not only to improve the segmentation accuracy but also to help the network optimization process, resulting in a faster convergence of the training [97].

A CNN architecture, called DenseNet, was introduced in [98], [99]. DenseNets are built from dense blocks and pooling operations, where each dense block is an iterative concatenation of previous feature maps. This architecture can be seen as an extension of ResNets [75], which performs the iterative summation of previous feature maps as shown in Fig. 2.3. However, this small modification has some interesting implications: (1) parameter efficiency, DenseNets are more efficient in the parameter usage; (2) implicit deep supervision, DenseNets perform deep supervision thanks to short paths to all feature maps in the architecture (similar to Deeply Supervised Networks [99]); and (3) feature reuse, all layers can easily access their preceding layers making it easy to reuse the information from previously computed feature maps. The characteristics of DenseNets make them a very good fit for semantic segmentation as they naturally induce skip connections and multi-scale supervision.

CNNs have revolutionized fingerprint recognition by offering a data-driven approach that achieves superior accuracy, robustness, and scalability compared to traditional techniques. As research in deep learning continues to advance, CNN-based fingerprint recognition systems are expected to become even more accurate and reliable, further enhancing their applicability in various real-world applications such as law enforcement, access control, and biometric authentication.

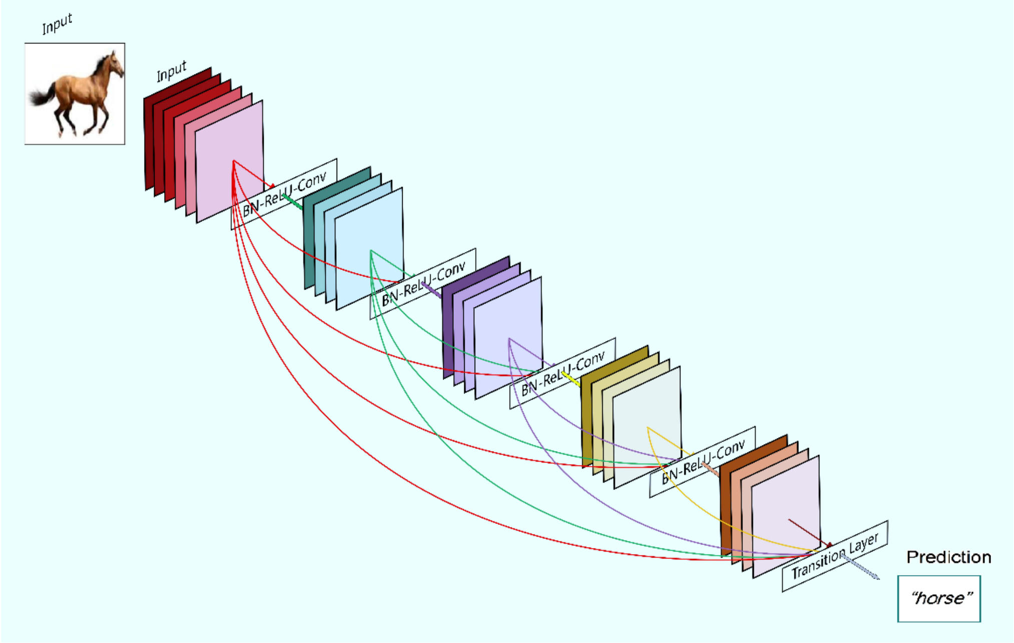


Figure 2.3: A dense block with 5 layers and growth rate 4.

2.3 Performance Metrics

A classification model serves as a mapping from instances to predicted classes [100]. Each instance is associated with a single element representing positive or negative class labels. Given a classifier and an instance, four potential outcomes exist. If a positive instance is correctly classified as positive, it's termed a true positive; if classified as negative, it's deemed a false negative. Similarly, if a negative instance is correctly classified as negative, it's a true negative; otherwise, if classified as positive, it's a false positive. By employing a classifier and a set of instances (the test set), a two-by-two confusion matrix (contingency table), as illustrated in

Table 2.1, can be constructed to depict the disposition of instances, forming the basis for various metrics.

The true positive rate (TP rate) and false positive rate (FP rate) can be calculated using the following equations:

$$TP\ Rate = \frac{Positives\ Correctly\ Classified}{Total\ Positives} \tag{2.3}$$

$$FP\ Rate = \frac{Positives\ Incorrectly\ Classified}{Total\ Negatives} \tag{2.4}$$

Table 2.1: Confusion Matrix for calculating hit rate for a classification model.

		Actual Class	
		Positive	Negative
Predicted Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

These rates are represented on a receiver operating characteristics (ROC) graph, with FP rate on the x-axis and TP rate on the y-axis. Each classifier generates a (FP rate, TP rate) pair, corresponding to a point in ROC space. The ROC curve is a fundamental tool in statistics and machine learning, particularly for evaluating the performance of classification models. It provides insights into how well a model distinguishes between positive and negative instances across different decision thresholds. In Fig. 2.4, each point on the curve corresponds to a different threshold applied to the model’s predicted probabilities. As the curve approaches the perfect classifier point, the model’s performance improves and a steeper curve towards that point indicates better discrimination. Conversely, deviations away from the perfect classifier indicate worse performance. A random guess typically results in a point falling along a diagonal line, known as the line of no-discrimination, stretching from the bottom-left to the top-right corners, irrespective of the positive and negative base rates [101]. A simple illustration of random guessing is akin to making decisions by flipping coins. As the sample size grows, the ROC point of a random classifier converges towards this diagonal line. In the scenario of a balanced coin, this convergence leads to the point (0.5, 0.5). ROC curve provides valuable insights for model selection and optimization, especially when dealing with imbalanced datasets or critical decision-making scenarios.

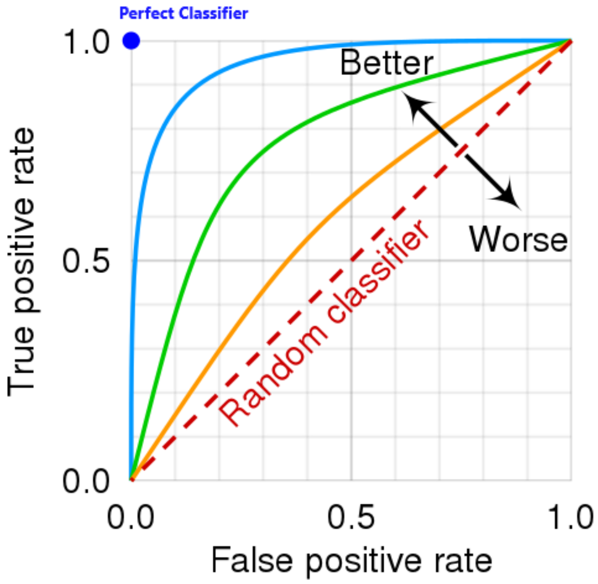


Figure 2.4: Basic ROC graph showing different schemes curves.

In fingerprint classification, a fingerprint that belongs to a class but isn't identified by the classifiers is categorized as a false negative. Given that fingerprint classification precedes various fingerprint applications such as fingerprint recognition, false negatives pose significant challenges as they lead to incorrect results in subsequent stages.

In addition to these metrics, there are other parameters for biometric framework verification:

- False Accept Rate (FAR) or False Matching Rate (FMR): Measures the likelihood of the system incorrectly matching input queries to non-matching templates in the database, indicating the percentage of invalid inputs that are erroneously accepted.
- False Reject Rate (FRR) or False Non-Matching (Negative Matching) Rate (FNMR): Measures the likelihood of the system failing to identify a match between the input query and a matching template in the database, quantifying the percentage of valid inputs that are incorrectly rejected.

- Equal Error Rate (EER) or Crossover Error Rate (CER): The rate at which FMR and FNMR are equal. EER can be easily obtained from the ROC curve, providing a quick means to compare the accuracy of devices with different ROC curves. Generally, the scheme with the lowest EER is considered the most accurate, derived from the point on the ROC curve where FMR and FNMR are equal.

2.4 Euclidean Distance

In mathematics and geometry, the Euclidean distance plays a fundamental role [102]. It measures the length of the straight-line segment connecting two points in Euclidean space. This space adheres to the axioms and postulates of Euclidean geometry, which we encounter in our everyday physical world. The Euclidean distance between two points **P** and **Q** in Euclidean space is precisely the length of the line segment that directly connects these points. It provides a straightforward way to quantify the spatial separation between them. To calculate the Euclidean distance, consider two points with Cartesian coordinates as showing in Fig 2.5:

- Point **P**: (p_1, p_2)
- Point **Q**: (q_1, q_2)

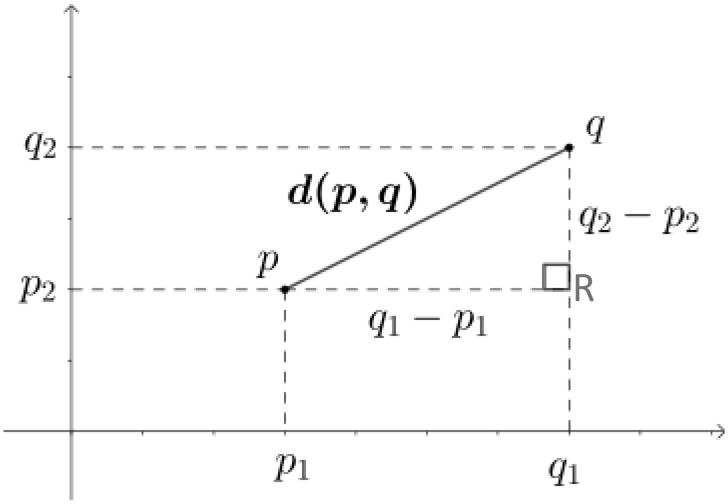


Figure 2. 5: Two-dimensional Euclidean distance.

The Euclidean distance formula is derived from the Pythagorean theorem. It is expressed as:

$$ED = \sqrt{((q_1 - p_1)^2 + (q_2 - p_2)^2)} \tag{2.5}$$

By constructing a triangle using the points **P** and **Q**. The hypotenuse of this triangle corresponds to the line segment connecting the two points. Consider the following steps:

1. Join points **P** and **Q** using a straight line.
2. Construct a right triangle with hypotenuse **PQ**.
3. Draw horizontal and vertical lines from **P** and **Q**, intersecting at point **R**.

Applying the Pythagorean theorem [103] to triangle **PQR**, we get:

$$\text{Hypotenuse}^2 = Base^2 + \text{Perpendicular}^2 \tag{2.6}$$

Substituting the coordinates:

$$PQ^2 \ = \ PR^2 \ + \ QR^2 \tag{2.7}$$

Therefore:

$$ED^2 \ = \ (q_1 - p_1)^2 + (q_2 - p_2)^2 \tag{2.8}$$

Taking the square root of both sides, we arrive at the Euclidean distance formula as in equation (2.5)

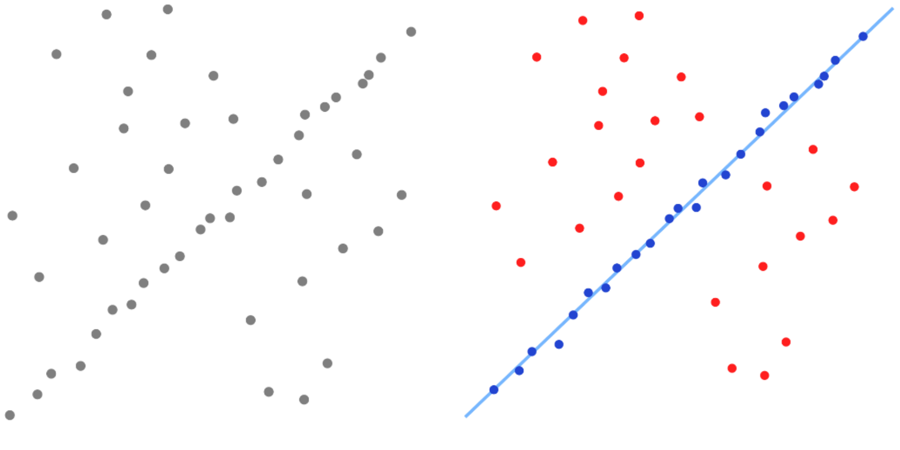
The Euclidean distance employs applications in various fields, such as data science, computer vision, geographic information systems and physics. Euclidean distance is a versatile metric that bridges geometry, mathematics, and practical applications. Its simplicity and effectiveness make it a basis in various scientific and engineering domains.

2.5 Random Sample Consensus (RANSAC) and Weighted Random Sample Consensus (WRANSAC)

The random sample consensus (RANSAC) algorithm is an iterative approach designed to estimate parameters of a mathematical model from a dataset containing outliers, ensuring that these outliers have minimal influence on the estimation process. Consequently, RANSAC can also be viewed as an outlier detection method [104]. This algorithm operates in a non-deterministic manner, meaning it yields a satisfactory result only with a certain probability, which increases as more iterations are allowed. RANSAC was initially applied to solve the location determination problem (LDP), aiming to identify points in space that project onto an image as a set of landmarks with known locations.

RANSAC employs repeated random sub-sampling [105]. It assumes that the dataset comprises "inliers," representing data explainable by a set of model parameters albeit subject to noise, and "outliers," which deviate from the model. Outliers may arise from extreme noise values, erroneous measurements, or incorrect hypotheses regarding data interpretation. Additionally, RANSAC assumes the existence of a procedure capable of estimating model parameters optimally when provided with a small set of inliers.

For instance, consider fitting a line in two dimensions to a dataset comprising both inliers and outliers, as shown in Fig. 2.6. While a standard least squares method may produce a line poorly fitted to the data due to its inclusion of outliers, RANSAC aims to exclude outliers and derive a linear model solely from the inliers. By fitting linear models to multiple random subsets of the data and selecting the model with the best fit to a subset, RANSAC mitigates the impact of outliers. The likelihood of algorithm success hinges on factors such as the proportion of inliers in the dataset and the selection of algorithm parameters.



(a) (b)

Figure 2.6: Data fitting using RANSAC. (a) A data set with many outliers for which a line has to be fitted. (b) Fitted line with RANSAC, where outliers have no influence on the result.

The RANSAC algorithm is a robust learning technique utilized to estimate model parameters through random sampling of observed data. Operating on datasets containing both inliers and outliers, RANSAC employs a voting scheme to identify the optimal fitting result. This voting process relies on two fundamental assumptions: noisy features will not consistently support any single model (indicating few outliers), and there are sufficient features to converge on a reliable model (indicating few missing data). RANSAC comprises two iterative steps:

i. Sample subset selection: A minimal subset of data items is randomly chosen from the input dataset and a fitting model with model parameters is computed using only the elements of this sample subset. Note that the size of the sample subset ensures determinability of the model parameters.
ii. Consensus set determination: The algorithm evaluates all data elements against the model instantiated by the estimated parameters from the first step and the data elements deviating from the model within a specified error threshold (defining the maximum deviation of inliers) are deemed outliers. The set of inliers obtained for the fitting model is referred to as the consensus set and the algorithm iteratively repeats these steps until a consensus set with a sufficient number of inliers is obtained.

The input to the RANSAC algorithm includes a set of observed data values, a model for fitting to the observations, and confidence parameters defining outliers. In detail, RANSAC achieves its objective by iteratively repeating the following steps:

1. Select a random subset of the original data, designated as the hypothetical inliers.
2. Fit a model to the set of hypothetical inliers.
3. Test all data against the fitted model, designating data points fitting the model well as the consensus set (i.e., inliers for the model).
4. Evaluate the quality of the estimated model based on the size of the consensus set.
5. Improve the model by re-estimating it using all members of the consensus set, refining the model fitting based on the fitting quality measure.
6. Repeat the procedure a fixed number of times until convergence to a sufficiently good model parameter set, with each iteration either rejecting a model due to insufficient consensus set size or refining a model with a larger consensus set.

The determination of the threshold value t for data point fitting to a model and the required number of inliers (data points fitted within t) to assert the model's good fit to the data is contingent upon the specific application requirements and dataset characteristics, often informed by experimental evaluation. However, the number of iterations (N) can be approximately derived as a function of the desired probability of success (ρ), as outlined below.

Let ρ denote the desired probability that the RANSAC algorithm yields at least one useful result upon execution. In an ideal scenario, RANSAC returns a successful result if, in some iteration, it exclusively selects inliers from the input data set when choosing n points for estimating model parameters. Let v represent the probability of selecting an inlier each time a single data point is chosen, roughly defined as:

$$v = \frac{(number\ of\ inliers\ in\ data)}{(total\ number\ of\ points\ in\ data)} \tag{2.9}$$

While v may not be precisely known beforehand due to the unknown number of inliers in the data prior to RANSAC execution, a rough estimate can suffice. Assuming n points needed for model estimation are selected independently (a rough assumption due to the reduction in available data points with each selection), the probability that all n points are inliers is v^n, while the probability of at least one outlier among the n points is $1 - v^n$, indicating potential estimation of a flawed model. The probability to the power of N iterations (the probability that the algorithm never selects a set of n points consisting entirely of inliers) is equivalent to $1 - \rho$ (the probability of unsuccessful model estimation). Thus, the relationship can be expressed as:

$$1 - \rho = (1 - v^n)^N \tag{2.10}$$

taking the logarithm of both sides' yields:

$$N = \frac{\log(1-\rho)}{\log(1-v^n)} \tag{2.11}$$

This derivation assumes independent selection of n data points, implying that a point selected once can be replaced and selected again within the same iteration. While this may not always be realistic, the derived value for N should be considered an upper limit in scenarios where points are selected without replacement.

WRANSAC, an extension of the renowned RANSAC algorithm, introduces a novel paradigm by integrating weighted considerations into the sampling and model evaluation phases [106]. This enhancement empowers WRANSAC to dynamically adjust its focus, prioritizing data points with higher reliability or relevance, thus fostering a more adaptable and robust model estimation process. In contrast to the conventional RANSAC, WRANSAC's weighted approach enables it to better accommodate scenarios where certain data points carry more significance, potentially leading to more accurate model fittings and reduced susceptibility to outliers. For instance, in computer vision applications, WRANSAC may assign higher weights to well-tracked features, thereby emphasizing their contribution to the model estimation.

WRANSAC unfolds through several key steps, starting with the assignment of weights to individual data points, often informed by prior knowledge, sensor confidence, or domain-specific considerations. Subsequently, the algorithm employs weighted sampling techniques, favoring data points with higher assigned weights during the random sampling process. During model evaluation, these weights play a pivotal role, influencing both the model's quality assessment and its subsequent refinement. By iteratively adjusting weights and refining the model, WRANSAC converges towards a robust solution, striking a harmonious balance between the influence of inliers and outliers, ultimately enhancing its resilience and adaptability in diverse data environments.

2.6 Local Topology Preserving Matching (LTPM) Algorithm

In the pore feature extraction and matching module, the final result of the pore matching segment is input to a refinement algorithm to remove any falsely matched pair of pores between the query and template fingerprint images. These refinement algorithms focus on detecting reliable correspondences, akin to a process of mismatch removal. Various algorithms, such as position-based and graph-based approaches, can be employed for this task. Position-based algorithms, such as RANSAC and WRANSCA, utilize geometric distances between points to differentiate correct matches from outliers, while graph-based algorithms leverage spatial relationships between points to enhance robustness against local deformation. In this study, we discuss the pairwise locality preserving matching (LTPM) algorithm [54] for refining pore correspondences, i.e., list of the matched pair of pores from the query and template fingerprints, known for its accurate and efficient mismatch detection capabilities. This approach is rooted in the locality preserving matching (LPM) algorithm [107].

The LPM algorithm revolves around the fundamental concept of leveraging neighbor relationships and topological similarities between points in template and query images to assess correspondence similarity. This algorithm operates on a correspondence set denoted as $S = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i and y_i represent coordinates of points from query and template images, undergoes comparison problem expressed as

$$p^* = arg_I \min C(p; S, \lambda) \tag{2.12}$$

where $p \in \{0,1\}^{1\times N}$ serves as an indicator vector, p_i = 1 denotes an inlier and p_i = 0 indicates an outlier. The parameter λ modulates the weight attributed to the corresponding part in equation (2.16) Here, C represents the cost function, N stands for the total number of correspondences, and I denotes the unknown inlier set. For each point x_i, the algorithm conducts independent searches for its K-nearest neighbors, denoted as N_{x_i}, along with their respective correspondences y_i and N_{y_i}. Subsequently, the cost C is computed based on the distance metrics $d(x_i, x_j)$ and $d(y_i, y_j)$, as delineated below.

$$d(x_i, x_j) = \begin{cases} 1, & x_j \in N_{x_i} \\ 0, & x_j \notin N_{x_i} \end{cases} \qquad d(y_i, y_j) = \begin{cases} 1, & y_j \in N_{y_i} \\ 0, & y_j \notin N_{y_i} \end{cases} \tag{2.13}$$

Utilizing the local topological structure, the algorithm enhances the calculation of similarity between correspondence pairs. For every pair (x_i, y_i), an edge is created with $v_i = y_i - x_i$ (see Fig. 2.7 (a)). Subsequently, the topological similarity between the two correspondences is computed using the following equation:

$$s(v_i, v_j) = \frac{\min\{|v_i|, |v_j|\}}{\max\{|v_i|, |v_j|\}} \frac{(v_i, v_j)}{|v_i||v_j|} \tag{2.14}$$

Moreover, the distance between these edges is:

$$d(v_i, v_j) = \begin{cases} 1, & s(v_i, v_j) \leq \tau \\ 0, & s(v_i, v_j) > \tau \end{cases} \tag{2.15}$$

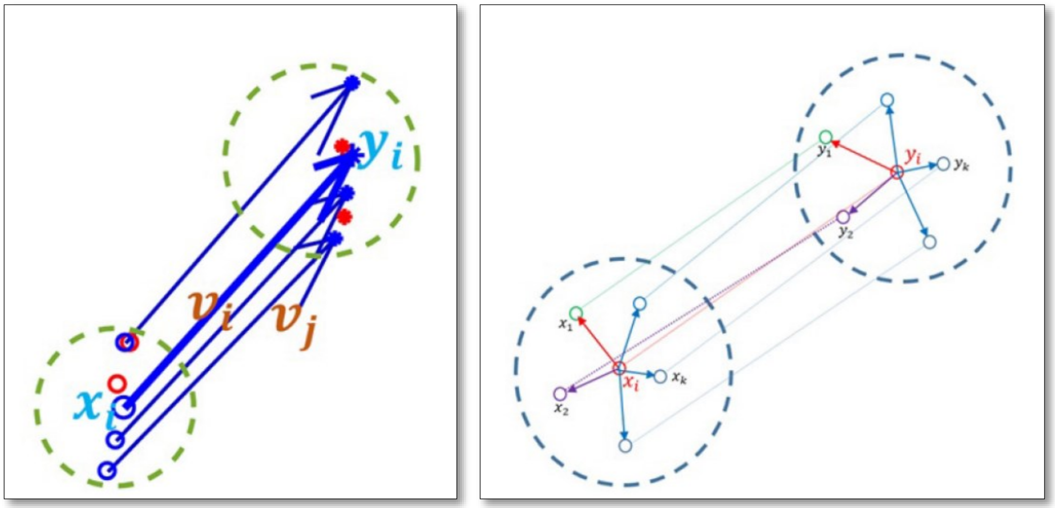
Lastly, the cost function is defined as

$$C(p) = \sum_{i=1}^{N} \frac{p_i}{K} \left(\sum_{j|x_j \in N_{x_i}} d(y_i, y_j) + \sum_{j|x_j \in N_{x_i}, y_j \in N_{y_i}} d(v_i, v_j) \right) + \lambda \left(N - \sum_{i=1}^{N} p_i\right) \tag{2.16}$$

This problem can be easily solved using a closed-form solution, requiring minimal computational complexity.

While the LPM algorithm has demonstrated effectiveness in removing mismatches, it suffers from a notable drawback. This algorithm constructs a displacement vector $v_i = y_i - x_i$ for each correspondence pair (x_i, y_i), and subsequently calculates the topological structure similarities between pairs based on differences in vector length and rotation. However, due to the non-rotational invariance of the displacement vector, the difference in length, computed as $\min\{|v_i|, |v_j|\} / \max\{|v_i|, |v_j|\}$, may fail to adequately account for small changes in length when significant translation exists between the images. Conversely, when rotation occurs between the test and template images, the difference in length between vectors becomes highly sensitive. To address this limitation, a rotational invariant local topology structure is employed to assess the similarities between correspondence pairs [54].

To mitigate these issues, consider a point x_i and its k-nearest neighbors $\{x_t, t = 1,2, \dots, k\}$, with corresponding pores denoted as y_i and $\{y_t, t = 1, 2, \dots, k\}$. For each corresponding pore (x_t, y_t), two directional edges are independently constructed from (x_i, y_i). This ensures a one-to-one correspondence between edges, as illustrated in Fig. 2.7 (b).



(a) (b)

Figure 2.7: LPM and LTPM algorithms description. (a) Local structure constructed by LPM algorithm. (b) LTPM algorithm.

Consider an edge consisting of x_i and its jth nearest neighbor x_j denoted as e_{ij}^x, with its corresponding edge e_{ij}^y. A vector ϑ_{ij} is then formulated for each edge pair to represent the rotation between them.

$$\vartheta_{ij} = \left[\cos\left(\theta_{ij}^x - \theta_{ij}^y\right).\sin\left(\theta_{ij}^x - \theta_{ij}^y\right)\right] \tag{2.17}$$

The similarity between corresponding pairs $\left(e_{ij}^x, e_{ij}^y\right)$ and $\left(e_{il}^x, e_{il}^y\right)$ is computed using the following equation:

$$S_{jl} = \vartheta_{ij}\vartheta_{il}^T = \frac{\min\{d_{ij,x}, d_{ij,y}\}}{\max\{d_{ij,x}, d_{ij,y}\}} \cdot \frac{\min\{d_{il,x}, d_{il,y}\}}{\max\{d_{il,x}, d_{il,y}\}} \tag{2.18}$$

where $d_{ij,x}$ and $d_{ij,y}$ represent the lengths of e_{ij}^x and e_{ij}^y, respectively. For each corresponding edge pair $\left(e_{ij}^x, e_{ij}^y\right)$, a cumulative comparison score is computed as

$$V_j = \sum_{l=1}^{k} S_{jl} \tag{2.19}$$

The corresponding edge pair with the highest cumulative score is designated as a datum:

$$r = arg \max_j V_j \tag{2.20}$$

This datum edge pair serves as a reference for evaluating the similarities of other edge pairs using equation (2.18). The similarity between the l^{th} correspondence is denoted as $S_{rl}, l = 1,2, \dots, k$. The similarity between each pair (x_i, y_i) can then be calculated as:

$$s_i = \frac{1}{k}\sum_{l=1}^{k} S_{rl} \tag{2.21}$$

Finally, the cost function is defined as

$$C(p) = \sum_{i=1}^{N} \frac{p_i}{K}\left(\sum_{j|x_j \in N_{x_i}} d\left(y_i, y_j\right) + \sum_{j|x_j \in N_{x_i}, y_j \in N_{y_i}} (1 - S_{rl})\right) + \lambda\left(N - \sum_{i=1}^{N} p_i\right) \tag{2.22}$$

The solution to the above equation is defined similar to LPM algorithm as

$$p_i = \begin{cases} 1, & c_i \leq \lambda \\ 0, & c_i > \lambda \end{cases} \tag{2.23}$$

2.7 Summary

In this chapter, fundamental components necessary for the development of fingerprint recognition systems have been briefly discussed. Convolutional neural networks, their architectures and basic elements such as convolution layers and activation functions have been introduced. Metrics for evaluating the performance of fingerprint recognition systems, such as FAR, FRR, ROC curve, and EER have been defined. The significance of Euclidean distance in feature comparison, particularly in fingerprint recognition, has been discussed. The techniques, such as RANSAC, WRANSAC, and LTPM, for refining a matched set of features have been described.

Chapter 3

Proposed Pore Detection Module

3.1 Introduction

In this chapter, a two-part scheme for pore detection is developed [108]. In the first part, a low-complexity deep neural network is designed to generate a highly representational pore intensity map by focusing on the connectivity of different hierarchical levels of the network feature extraction and learning these features locally in a residual framework and globally through a concatenative feature fusion. In the second part of the scheme, a method is developed for the detection of the pore centroids from the pore intensity maps obtained in the first part by making efficient use of the existing knowledge of the properties of fingerprint pores.

3.2 Proposed Scheme

Fig. 3.1 shows the block diagram of a neural network-based pore detection scheme (hybrid scheme) depicting its two-part. The first part is the pore feature extraction part using a convolutional neural network, and the second part is a postprocessing part that determines the centroids of the true pores. The input to the pore detection system is a grey-level fingerprint image $\boldsymbol{X_i}$, and the output is a map $\boldsymbol{X_t}$ showing the centroids of the detected pores. The first part of the pore detection scheme extracts the features at various levels using a neural network from the input fingerprint image $\boldsymbol{X_i}$ and then uses them to construct a single grey-level pore intensity map $\boldsymbol{X_o}$. The postprocessing part of the pore detection scheme is divided into two steps, Step 1 and Step 2. Step 1 is a pore centroid detection step, which receives the pore intensity map $\boldsymbol{X_o}$ as input and first transforms it into a binary pore map in which a value of 1 indicates that the pixel in question belongs to one of the pore regions, whereas a value of zero indicates that it does not. Using this binary pore map, the centroid of each candidate pore is determined. Thus, the output of step 1 of the second part of the scheme is the map $\boldsymbol{X_c}$ giving the coordinates of the centroids of each candidate pores. Step 2 is a pore refinement scheme, in which some of the candidate pores falsely detected as pores are discarded to yield a pore map $\boldsymbol{X_t}$ representing the coordinates of the centroids of the detected true pores. We next develop the proposed neural network-based scheme in the framework of the block diagram of Fig. 3.1.

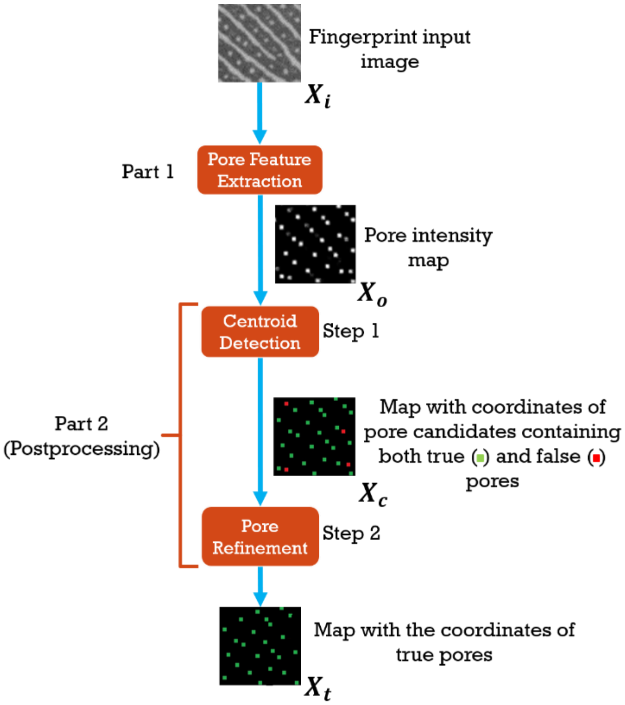


Figure 3.1: General structure of a neural network-based pore detection scheme. The approach takes a fingerprint image as an input and detects the true pore coordinates.

3.2.1 Part 1: Pore Feature Extraction

In this section, a CNN model for pore feature extraction is proposed. The design of this model aims at developing a feature extraction process that is robust enough to extract a very rich set of pore features in a computationally efficient manner. This is achieved through hierarchical feature extraction locally using convolutional layers in a residual framework and through their concatenative fusion globally.

Our objective of developing a very lightweight network with the capability of providing a rich set of pore features leading to a highly representational pore intensity map is based on the following principles.

- The network is designed to have three stages, stages 1 to 3, that produce, respectively, low, middle and high levels of hierarchal pore features.

- A concatenative fusion of the three levels of features must produce a very rich set of pore features for the construction of a highly representational pore intensity map.
- Recognition that the generation of a pore intensity map is essentially a classification problem, in which each pixel is classified to belong or not to belong to a pore region. A classification problem can be better served in a convolutional network by using depthwise convolution rather than the standard convolution.
- A major consequential advantage of depthwise convolution is that its use leads to the development of a CNN architecture with a dramatically reduced complexity.

The proposed network model consists of four stages, as shown in Fig. 3.2. The first three stages carry out the processes of feature extraction and their fusion using a local skip connection. The last stage performs a global feature fusion of the features extracted locally at three hierarchal levels by the first three stages. Each of the three feature extraction stages consists of two convolution layers, and a module to residually learn the features locally. The skip connections used in each of the three stages curtail the gradient vanishing problem of a deep network [75], [76], [77] leading to better training of the network parameters, and hence, generation and extraction of more accurate features. The features produced by stages 1, 2 and 3 can be considered to be, respectively, the low, middle and high level pore features in view of the hierarchical levels of the convolutional layers used in these stages.

In order to achieve sufficient diversity among the extracted features of different channels and yet to keep the complexity low, we apply eight different 3×3 kernels on an input patch $\boldsymbol{X_{ip}}$ of the input image $\boldsymbol{X_i}$ to obtain eight distinct channels in the first layer of stage 1. Each of the eight resulting feature maps of the first layer of this stage is operated on by another 3×3 kernel. Thus, the second layer of this stage also employs eight different kernels to produce eight distinct maps. The additive residual fusion of the corresponding maps of the two layers then yield the output feature maps $\boldsymbol{X_1}$ of the first stage.

In order to provide further diversity in the extracted features while keeping the complexity of the network low, we doubled the number of filters, in each of the other two feature extraction stages from that in the previous stage. Specifically, in the first layer of stage 2, we apply two 3×3 kernels on each of the eight maps of $\boldsymbol{X_1}$ resulting from stage 1 and one 3×3 kernel in the second layer to each of the extracted maps from the first layer of this stage. Thus, each of the two layers of stage 2 uses 16 3×3 different kernels. Similarly, in stage 3, two kernels are applied to each of the 16 maps of $\boldsymbol{X_2}$ and one 3×3 kernel to each of the extracted maps from its first layer. Hence, each of the two layers of stage 3 employs a total of 32 3×3 different kernels, a number that is twice that used by the layers in stage 2. In the first three stages, batch normalization (BN) [109] and non-linear activation function (ReLU) [110] are performed after each convolution operation in order to normalize the distribution of the features produced by the previous layer.

The main function of the last stage, stage 4, is two-fold. First, it should suitably combine the feature maps produced by the first three stages so that the significance of the individual features is taken into account to produce a pore map. Second, the values of the individual features should provide a reasonable indication of its likelihood of belonging to a pore. The feature maps produced by the different feature extraction stages are comparatively more diverse than the feature maps produced locally by a single stage. Therefore, a weighted combination of the feature maps $\boldsymbol{X_1}$, $\boldsymbol{X_2}$ and $\boldsymbol{X_3}$ should be carried out. This purpose is effectively served by performing a point-wise convolution of the feature map $\boldsymbol{X_{C_i}} = (\boldsymbol{X_1}\ \ \boldsymbol{X_2}\ \ \boldsymbol{X_3}\ )^{\boldsymbol{T}}$, since this type of convolution provides a mechanism of summation of the corresponding pixels in the maps of $\boldsymbol{X_{C_i}}$ with weights that are learnable. Thus, in our proposed scheme, a point-wise convolution is performed on $\boldsymbol{X_{C_i}}$ using a single filter of kernel size 1×1×56 pixels in order to produce a single map $\boldsymbol{X_{C_o}}$. Now, as mentioned earlier, each pixel value of the network output patch $\boldsymbol{X_{op}}$ must be indicative of its likelihood of belonging to a pore. Therefore, a pixel with a large positive value can be regarded to belong to a pore with high certainty, whereas a pixel with a negative value but a large magnitude can be regarded to belong to a pore with very low certainty. On the other hand, the pixel values of $\boldsymbol{X_{C_o}}$in the neighborhood of zero need some amplification for further consideration of their likelihood of belonging to a pore. Essentially, this last step of stage 4 in going from $\boldsymbol{X_{C_o}}$to $\boldsymbol{X_{op}}$ is a classification problem in which the likelihood of a pixel belonging to a pore is determined. In our proposed scheme, we achieve this goal by applying a sigmoid activation function [32] on $\boldsymbol{X_{C_o}}$ to yield $\boldsymbol{X_{op}}$. Thus, each pixel value of $\boldsymbol{X_{op}}$ lies in the range 0 to 1 indicating its likelihood of belonging to a pore. Once the pore intensity maps $\boldsymbol{X_{op}}$'s for all the patches of $\boldsymbol{X_i}$ are determined, the network finally recomposes them into a single pore intensity map $\boldsymbol{X_o}$ corresponding to the input image $\boldsymbol{X_i}$.

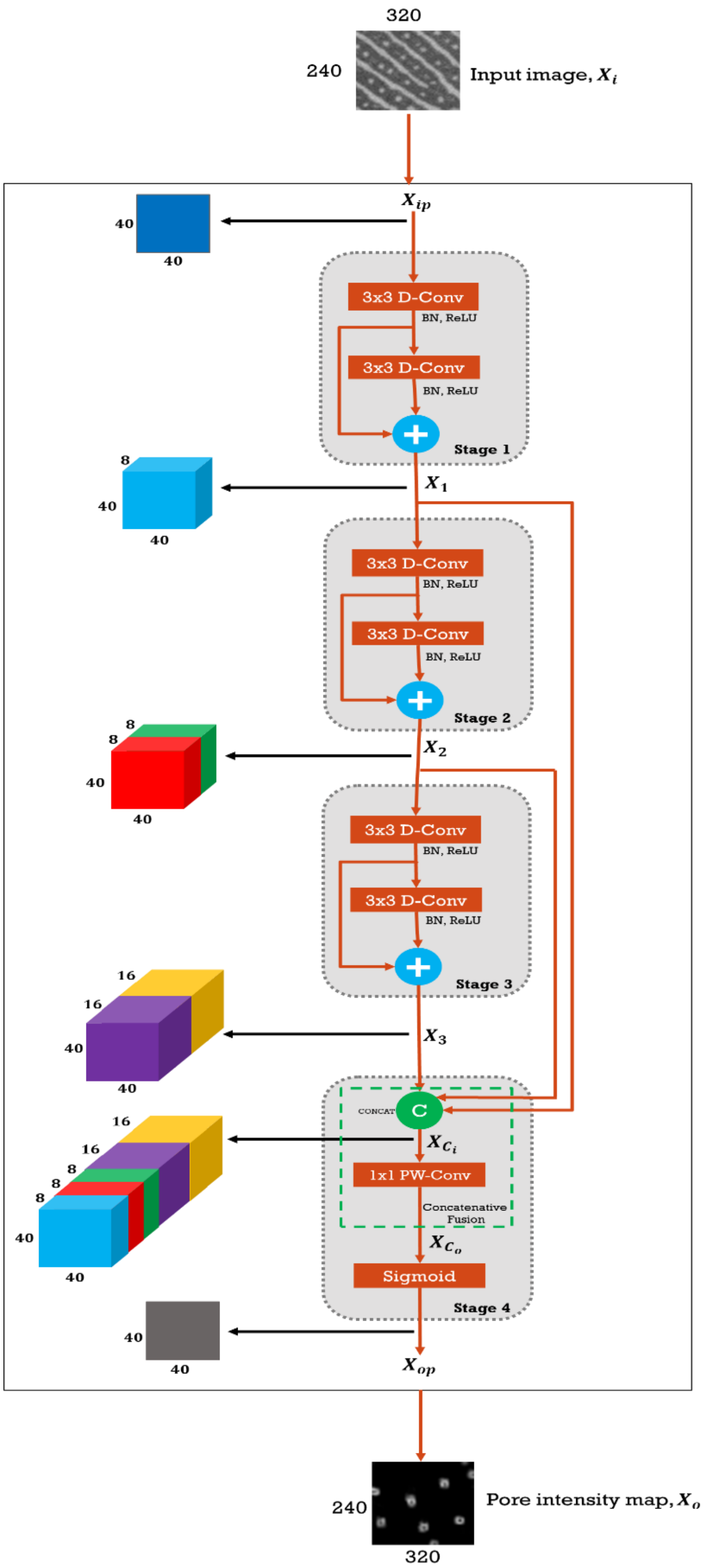


Figure 3.2: Proposed network architecture for pore detection.

3.2.1 Part 2: Postprocessing

In this section, a method is presented for accurately determining the coordinates of all the true pores from the pore intensity map $\boldsymbol{X_o}$ obtained from the proposed pore feature extraction network presented in the previous section, while attempting to discard the false pores. This postprocessing part of the proposed pore detection scheme consists of two steps as shown in the block diagram of Fig. 3.3. Step 1 consists of an image binarization module that transforms the pore intensity map $\boldsymbol{X_o}$ into a binary map $\boldsymbol{X_b}$ and a pore determination module that obtains a candidate pore centroid map $\boldsymbol{X_c}$ from $\boldsymbol{X_b}$. Step 2 consists of a pore refinement process that identifies the true pores and removes false pores from $\boldsymbol{X_c}$ leading to a final pore map $\boldsymbol{X_t}$ containing the centroids of the true pores in the fingerprint image.

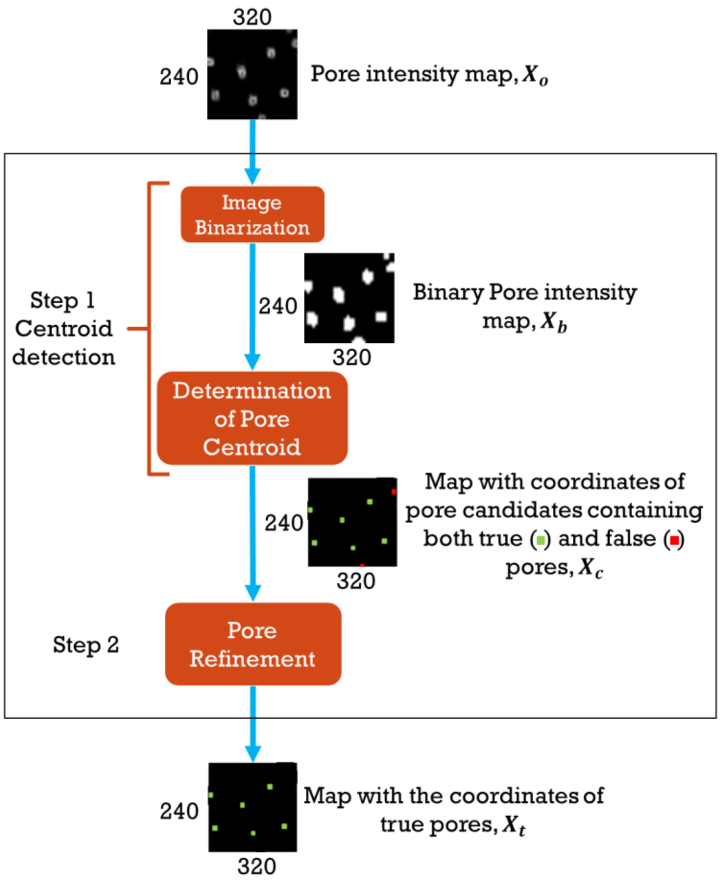

320
240
Pore intensity map, X_o
Image Binarization
Step 1 Centroid detection
240
Binary Pore intensity map, X_b
320
Determination of Pore Centroid
240
Map with coordinates of pore candidates containing both true (🟩) and false (🟥) pores, X_c
320
Step 2
Pore Refinement
240
Map with the coordinates of true pores, X_t
320

Figure 3.3: Structure of the proposed postprocessing part.

Step 1: Centroid Detection

A. Image Binarization

Since the pore intensity map $\boldsymbol{X_o}$ gives only a measure of the likelihood of its individual pixels to belong to a pore, it is necessary to differentiate more clearly whether or not a pixel of $\boldsymbol{X_o}$ belongs to a pore region by assigning it a distinct binary value. Therefore, in order to obtain a binary map $\boldsymbol{X_b}$ from the pore intensity map $\boldsymbol{X_o}$, we need a suitable threshold T so that a pixel with a value equal to or greater than T could be classified as a candidate belonging to a pore region by assigning them a value of 1. On the other hand, if a pixel in $\boldsymbol{X_o}$ has a value less than T, it is assigned a value of 0 and, therefore, regarded not to belong to a pore region.

First of all, it is to be noted that the intensity values in $\boldsymbol{X_o}$ are directly related to the intensity distributions of the various pores within a fingerprint image $\boldsymbol{X_i}$. Even though the majority of the pores in the input image $\boldsymbol{X_i}$ have the same shape and intensity pattern, there are always some pores whose shapes and intensity values may differ, and hence, the intensity patterns of such pores may be different from those of a typical normal pore. The distribution of the pixel values in $\boldsymbol{X_o}$ is necessarily affected by such a nonuniformity of the pore shapes and intensity values in $\boldsymbol{X_i}$. Thus, if a fixed threshold T is chosen for the entire map $\boldsymbol{X_o}$ to convert it to a binary map $\boldsymbol{X_b}$, it may result in classifying some of the pixels of $\boldsymbol{X_o}$ to be falsely classified to belong or not to belong to a pore region. Hence, a method for determining the threshold must take into account the non-identically distributed nature of the pore distribution across the pore intensity map $\boldsymbol{X_o}$. In the following, we describe a scheme for determining a locally adaptive threshold T_l to binarize the pore intensity map $\boldsymbol{X_o}$, that is, it is adaptive from one window to another window of $\boldsymbol{X_o}$. When this threshold T_l is used for the window l of $\boldsymbol{X_o}$, then $\boldsymbol{X_o}$ is converted to a binary map $\boldsymbol{X_b}$.

Window-based locally adaptive binarization methods, which compute a threshold value for all the pixels within a given window, based on the information contained on the pixels value within the window, exist for applications other than pore detection. The Niblack algorithm [111] has been successfully used to determine a locally adaptive threshold T_l to identify the region of an object in document images. In this algorithm, the mean m_l and the standard deviation σ_l in the window l are used to determine a local value for the threshold T_l to be used for all the pixels within the window l, as

$$T_l = m_l + \beta\sigma_l \tag{3.1}$$

with the parameters m_l and σ_l computed as

$$m_l = \frac{1}{N_l}\sum_{k=1}^{N_l} p_k \tag{3.2}$$

$$\sigma_l = \sqrt{\sum_{k=1}^{N_l}(p_k - m_l)^2 \Big/ N_l} \tag{3.3}$$

where p_k is the gray value of the k^{th} pixel in the l^{th} window of $\boldsymbol{X_o}$, N_l is the number of pixels in the l^{th} window, and β, called the Niblack factor, is a parameter that controls the effect of the local standard deviation in determining the boundary of an object. In [112], the Niblack factor was also made locally adaptive by selecting its value using the expression given by

$$\beta_l = -\frac{0.3\big((m_g \times \sigma_g) - (m_l \times \sigma_l)\big)}{\max(m_g \times \sigma_g\, , m_l \times \sigma_l)} \tag{3.4}$$

where $m_g\ and\ \sigma_g$ are the global mean and standard deviation of the entire map. Thus, the Niblack factor β_l is made to change from window to window but it remains fixed within a window. For the purpose of transforming the pore intensity map $\boldsymbol{X_o}$ to a binary map $\boldsymbol{X_b}$, we employ the adaptive threshold T_l given by (3.1) with the adaptive Niblack factor β_l of (3.4). For the sake of simplicity, we use windows of fixed size N for the entire pore intensity map $\boldsymbol{X_o}$, i.e., $N_l = N$.

The intensity and the pore shapes in a typical fingerprint image vary more globally than locally. Therefore, choosing a large window size would fail to capture this global variation in the fingerprint image. On the other hand, choosing a small size window would result in classifying the regions of the fingerprint image with noise as pore regions. Also, a small window size would adversely impact the processing time. Hence, the window size should be chosen suitably.

B. Determination of Centroids of Candidate pores

A single pore belongs to a specific region in the fingerprint image. The pixels belonging to this region cannot be disconnected. Therefore, typically a region of the binary map $\boldsymbol{X_b}$ that has, say, N_p pixels with the values of 1 and are connected can be considered to be a single pore region. However, it should be pointed out that this definition of pore region is not absolutely accurate, since for instance, if N_p is too small, the region may or may not considered to be a pore region, in view of the fact that a connected region of $\boldsymbol{X_b}$ with too few pixels could possibly be noise. On the other hand, if N_p is too large, it may be a situation where the region in question may be a region corresponding to two or more pores merged together. As to how such regions of the binary map are dealt with will be discussed in the next subsection. For the time being, we will consider a set of pixels of $\boldsymbol{X_b}$ that have a binary value of 1 and are connected to represent the region of a single pore.

Detection of a pore means finding the coordinates of a pixel in the corresponding pore region that can be used to describe the physical location of the pore. In previous works, pore centroids have been used to describe the locations of the pores. The centroid of a pore has been computed using only the coordinates of the pixels that belong to the pore region. However, the human visual system while determining the location of a pore focuses not only on the geometric center of the pore but it is also influenced by the intensity variation of the pixels in the pore region. Hence, the pore centroid is not a very accurate representation of the pore location. Consequently, in our scheme, the pore location is represented by its weighted centroid. We compute the weighted centroid by using the physical geometry of the pore region defined as a set of pixels that have a value of 1 and are connected in the binary map $\boldsymbol{X_b}$ as well as the pore intensity information of the underlying pixels from the fingerprint image itself. Accordingly, the weighted centroid (X, Y) of a candidate pore is computed as

$$(X, Y) = \left(\frac{\sum_{v=1}^{v=N_p} (X_v \times P_v)}{\sum_{v=1}^{v=N_p} P_v} \quad , \quad \frac{\sum_{v=1}^{v=N_p} (Y_v \times P_v)}{\sum_{v=1}^{v=N_p} P_v} \right) \tag{3.5}$$

where(X_v, Y_v) represents the coordinates of the v^{th} pixel in the pore region of the candidate pore from the binary map $\boldsymbol{X_b}$, P_v is the pixel gray value of the corresponding pixel in the input image $\boldsymbol{X_i}$, and N_p is the number of pixels within the pore region of the candidate pore. This process of determining the weighted centroids of the candidate pores results in the map $\boldsymbol{X_c}$ which consists of the locations of the detected candidate pores.

Step 2: Pore Refinement Process

In principle, one would expect that a fingerprint image has as many pores as the number of pore centroids in the map $\boldsymbol{X_c}$. However, in practice each centroid in $\boldsymbol{X_c}$ does not necessarily represent a true pore. In order to distinguish between true and false pores, we first make certain observations on the true pores in ground truth fingerprint images that will help in distinguishing a true pore from a false one.

The first observation is that a true pore has a distance from its nearest neighbor which varies from pore to pore. Fig. 3.4 shows an example of a ground truth fingerprint image. In this figure, the nearest neighbor of pore 1 is pore 2 and that for pore 3 is pore 4. It is seen from this example that the distance between 1 and 2 is different from that between 3 and 4. Fig. 3.5 shows the histograms of the frequency (numbers) of pores whose Euclidian distance from its nearest neighbor falls in a given range in all the 30 ground truth images of the PolyU High-Resolution Fingerprint database [35]. For example, there are 1,287 pores in all the fingerprint images in this database that have Euclidian distances that lie in the range [11,12) from their nearest neighbors. It is seen from Fig. 3.5 that in none of the fingerprint images in this database there is a pore with a Euclidian distance less than 4.12 from its nearest neighbor. We denote this minimum distance by d_{min}. Hence, in a weighted centroid map $\boldsymbol{X_c}$, if a pair of neighboring pores is detected with Euclidian distance less than d_{min}, both the pores in the pair cannot be simultaneously regarded to be true pores. Note that the conclusion of d_{min}= 4.12 is reached based on 12,767 pores contained in the images of the PolyU High-Resolution Fingerprint database. We believe that the same conclusion would be valid for the pores in the fingerprint images from any other dataset with the value of d_{min} multiplied by a factor equal to the ratio between the resolution level of the images in the dataset considered in our study and that of the images in the other dataset.

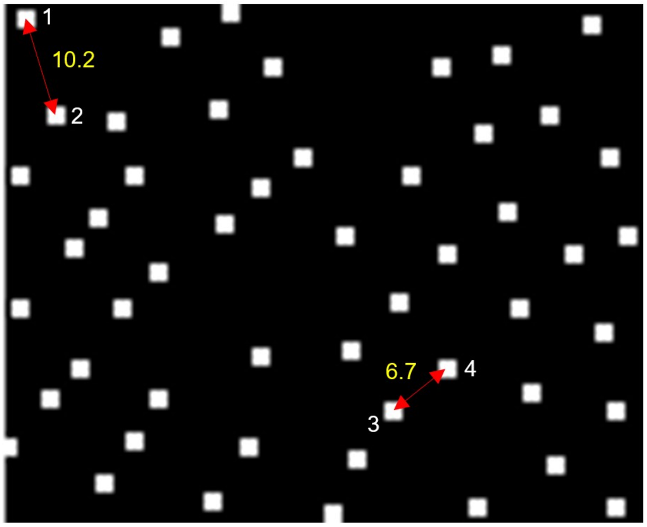


Figure 3.4: Two different pairs of true pores and their corresponding Euclidian distance value in the ground truth fingerprint image.

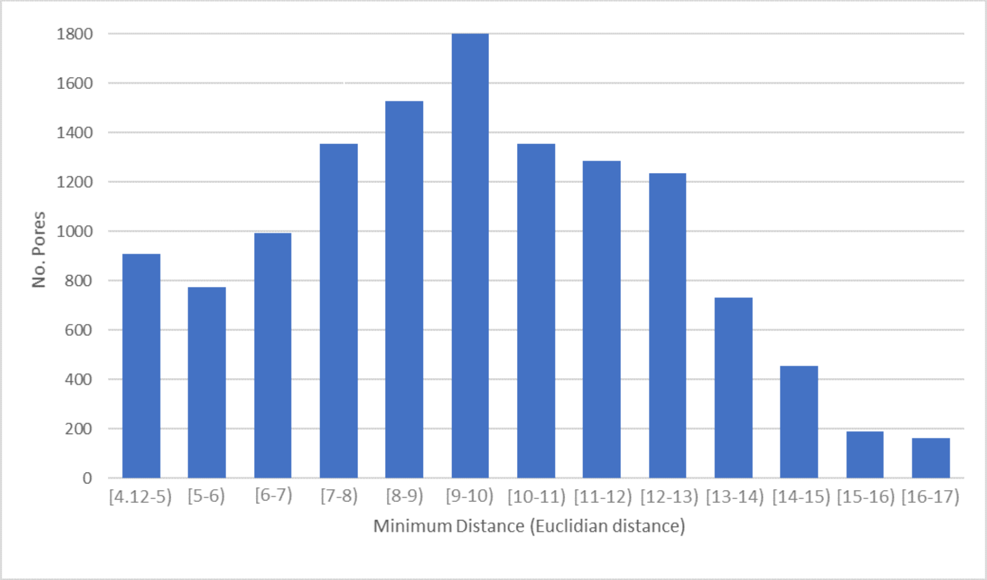
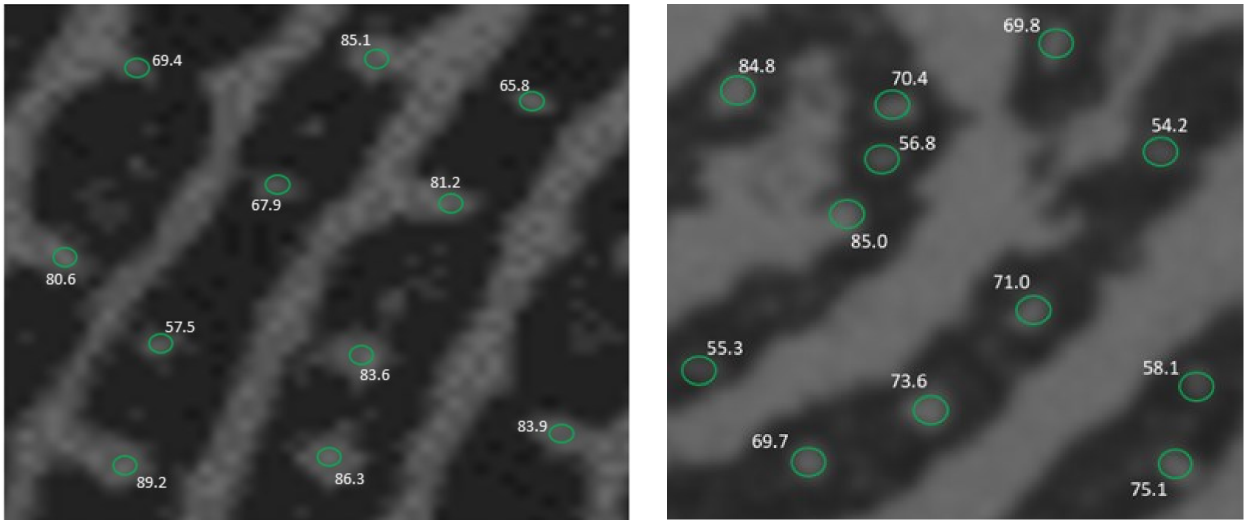


Figure 3.5: A histogram of the minimum distance between the two nearest pores in all fingerprint images of PolyU High-Resolution Fingerprint database.

The second observation is that, in different windows of a fingerprint image, there could be significant differences in the average intensities of their pores as well as in the variation of the intensities of the pores from the average intensity of the pores within the window. Fig. 3.6 shows two windows each of size 40×40 of a fingerprint image. It is seen from the fingerprint window in Fig. 3.6 (a) that the mean intensity of its pores is 77.3 and the variation of the pore intensities from this mean, represented by the standard deviation, is large. On the other hand, it is seen from the fingerprint window of Fig. 3.6 (b) that the mean intensity of its pores is smaller, and the variation of the pore intensities is also larger.

Both of the above observations could be used to distinguish true and false pores in a fingerprint image, and therefore, the weighted centroid map $\boldsymbol{X_c}$ could be refined by eliminating from it some of the centroids that correspond to false pores. The first observation that gives rise to the parameter d_{min} can be used to require the Euclidian distance between a pair of centroids in $\boldsymbol{X_c}$ to have a minimum value d_{min} for both the pores in the pair to be true pores. Based on the second observation, the mean and standard deviation values of the pore intensities in a given window in $\boldsymbol{X_c}$ have values that vary from window to window. Therefore, if the intensity of a pore in any given window has a value that is much smaller than the mean intensity of the pores in the window, then such a pore cannot be regarded to be a true pore, and hence, its centroid must be removed from $\boldsymbol{X_c}$. For the purpose of deciding whether the intensity of the pore is much smaller than the mean intensity of the pores within the window, its local standard deviation can be used. We now develop a criterion for each of the two observations to use it for refining $\boldsymbol{X_c}$.



(a) Mean = 77.3, STD = 9.85

(b) Mean = 65.9, STD = 10.1

Figure 3.6: Input fingerprint images with the intensity values of the true pores from the ground truth.

A. Procedure for Identifying and Removing False Pores From X_c using the d_{min} Criterion

In this subsection, we develop a systematic procedure for identifying as many false pores as possible in the pore map $\mathbf{X_c}$ based on the minimum distance criterion between a pair of pores. We describe our procedure through an illustrative example. For this purpose, we make use of a manually constructed pore map $\mathbf{X_c}$ rather than a pore map corresponding to a natural fingerprint image so that all the different possibilities of pores can be illustrated through a single example. Fig. 3.7 (a) shows one such pore map $\mathbf{X_c}$ with 16 centroids numbered 1 to 16.

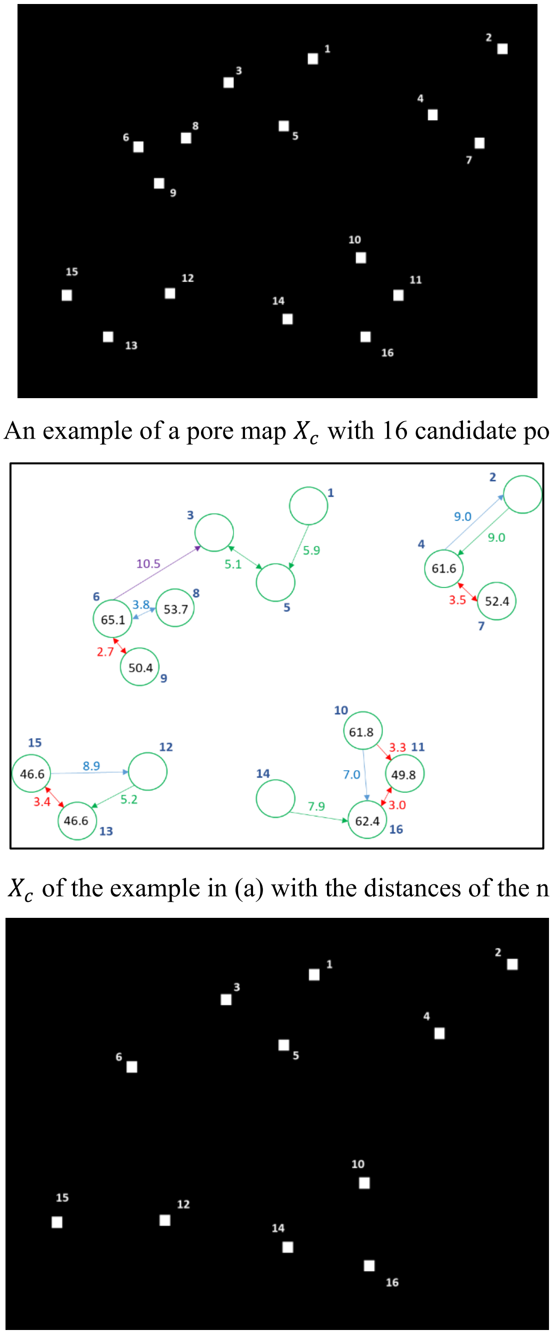
Our objective is to construct from this pore map $\mathbf{X_c}$ a refined pore map $\mathbf{X}_{\boldsymbol{d_{min}}}$(Fig. 3.7(c)) from which all the candidate pores in $\mathbf{X_c}$ that do not satisfy the minimum distance requirement are removed. Initially $\mathbf{X}_{\boldsymbol{d_{min}}}$is set to be $\mathbf{X_c}$. Then, starting from pore 1, we determine its Euclidian distance *ED (1, j)* to all the other pores j (j=2,…,16) and choose the pore that has the minimum distance from pore 1 as its nearest neighbor using the centroid values Pc_k (k=1,..,16). In the example considered, the nearest neighbor of pore 1 is found to be pore 5. This is shown in Fig. 3.7 (b) by connecting pore 1 to pore 5 with a green arrow directing from the former pore to the latter with the minimum distance of 5.9 indicated on the arrow. Since *ED (1,5)* between these two pores is larger than $d_{min} = 4.12$, both these pores are retained in $\mathbf{X}_{d_{min}}$.

The same procedure is followed for pores 2 and 3 to find their nearest neighbors, pores 4 and 5, with *ED (2,4) = 9* and *ED (3,5) =5.1.* It is noted that these pairs of pores also do not violate the minimum distance criterion, and hence, they are also retained in $\mathbf{X}_{d_{min}}$. For the next pore, i.e., pore 4, the nearest neighbor is found to be pore 7 with *ED (4,7) = 3.5,* which is less than d_{min}. Therefore, one of these pores in this pair must be removed from $\mathbf{X}_{d_{min}}$. In order to decide as to which pore in the pair has to be removed, we determine their pore intensities γ_4 and γ_7. The intensity of a pore j is found to be the average of the intensities of all the pixels in a 3×3 window centered at the centroid of pore j in the original fingerprint image $\mathbf{X}_i$ corresponding to $\mathbf{X}_c$. The intensity values γ_4 and γ_7 for pores 4 and 7 are, respectively, found to be 61.6 and 52.4, as indicated in Fig. 3.7 (b). Since γ_7 is less than γ_4, pore 7 is removed from $\mathbf{X}_{d_{min}}$ of Fig. 3.7 (c).

As pore 7 was the nearest neighbor of pore 4 and it has been removed, we need to find a new nearest neighbor of this latter pore. The new nearest neighbor of pore 4 is found to be pore 2 with *ED (4,2) =9* indicated by a blue arrow in Fig. 3.7 (b). Since *ED (4,2)* $>d_{min}$, pore 4 is still retained in $\mathbf{X}_{d_{min}}$. Next, since pore 5 does not violate the minimum distance criterion, it is retained in $\mathbf{X}_{d_{min}}$. As for pore 6, its nearest neighbor is pore 9 with *ED (6,9) = 2.7* $< d_{min}$. Since $\gamma_6 = 65.1$ and $\gamma_9 = 50.4$, pore 9 is removed from $\mathbf{X}_{d_{min}}$ and the new nearest neighbor of pore 6 is found to be pore 8. However, since *ED (6,8) = 3.8* is also less than d_{min}, the pore in the pair with the lower intensity, i.e., pore 8, is removed from $\mathbf{X}_{d_{min}}$. Thus, we find the third nearest neighbor of pore 6, which is pore 3 with *ED (6,3) =10.5* $>d_{min}$, indicated by a purple arrow, and consequently, both pores 6 and 3 are retained in $\mathbf{X}_{d_{min}}$.

Since pores 7, 8 and 9 are already removed from $\mathbf{X}_{d_{min}}$, next, we consider pore 10. For this pore, the nearest neighbor is pore 11 with *ED (10,11) =3.3* $<d_{min}$, and since $\gamma_{11} < \gamma_{10}$, pore 11 is removed from $\mathbf{X}_{d_{min}}$. Thus, we find the next nearest neighbor of pore 10, which is pore 16 with *ED (10,16) =7* $>d_{min}$, and consequently, both pores 10 and 16 are retained in $\mathbf{X}_{d_{min}}$. For the next pore, i.e., pore 12, the nearest neighbor is pore 13 with *ED (12,13) =5.2* $>d_{min}$, and thus, it is retained in $\mathbf{X}_{d_{min}}$. Next, the nearest neighbor of pore 13 is found to be pore 15 with *ED (13,15) =3.4* $<d_{min}$, thus, the pair of pores in question violates the minimum distance criterion. However, since in this case $\gamma_{13} = \gamma_{15} = 46.6$, we remove the first pore in the pair, i.e., pore 13,

from $\mathbf{X}_{d_{min}}$. The last three pores, i.e., pores 14, 15 and 16, as seen from Fig. 3.7 (b), do not violate the minimum distance criterion, and hence, are retained in $\mathbf{X}_{d_{min}}$. After taking into consideration all the pores in Fig. 3.7 (b) and removing all those that violate the minimum distance criterion, $\mathbf{X}_{d_{min}}$ shown in Fig. 3.7 (c) is the final composition of the refined pore map.



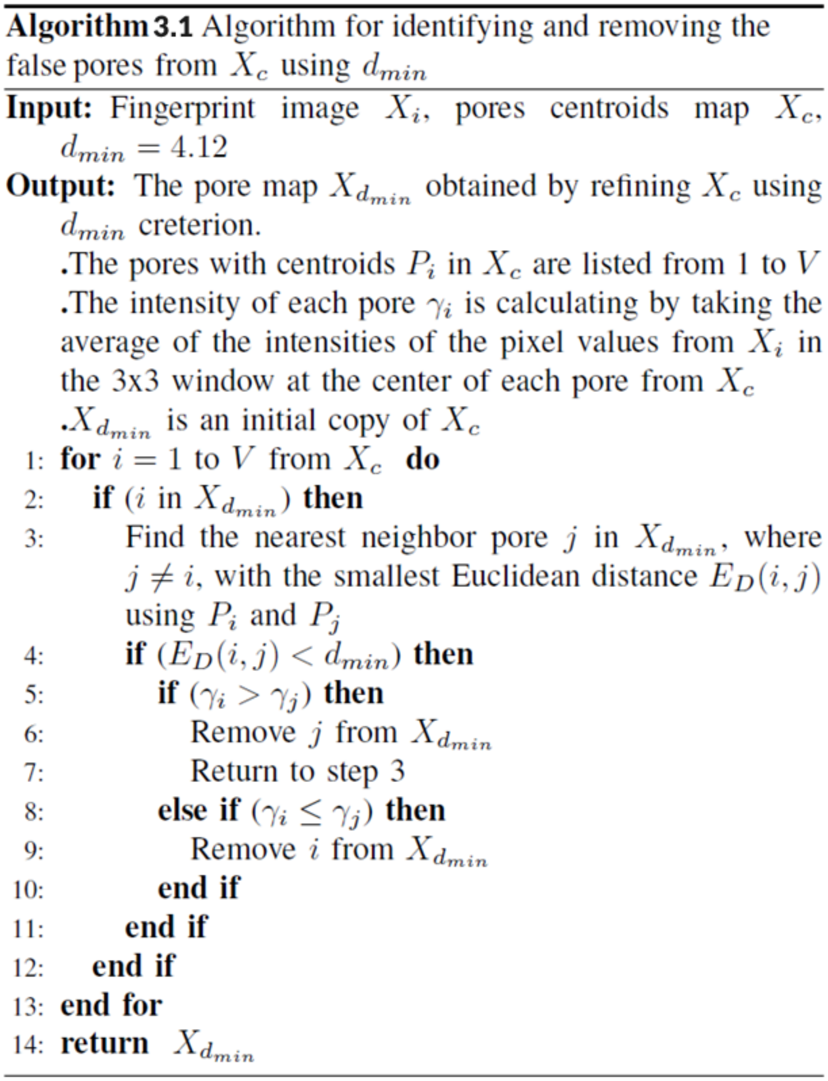
(a) An example of a pore map X_c with 16 candidate pores.

(b) Pore graph corresponding to X_c of the example in (a) with the distances of the nearest neighbors specified.

(c) Pore map $X_{d_{min}}$ corresponding to the pore map X_c in (a) after applying the d_{min} criterion.

Figure 3.7: Illustrations of the proposed refinement of the first criterion.

The procedure for identifying the false pores by applying the minimum distance criterion and removing the corresponding centroids from $\mathbf{X_c}$ is given in Algorithm 3.1. This algorithm presents our procedure for identifying and removing the false pores from $\mathbf{X_c}$ by applying thed_{min} criterion. The input to this algorithm is $\mathbf{X_i}$, $\mathbf{X_c}$ and dmin, and the output is the refined pore map $\mathbf{X}_{\boldsymbol{d_{min}}}$ in which the centroids in $\mathbf{X_c}$ corresponding to the pores not satisfying the d_{min} criterion has been removed.



Algorithm 3.1 Algorithm for identifying and removing the false pores from X_c using d_{min}

```
Input: Fingerprint image $X_i$, pores centroids map $X_c$,
    $d_{min} = 4.12$
Output: The pore map $X_{d_{min}}$ obtained by refining $X_c$ using
    $d_{min}$ creterion.
    .The pores with centroids $P_i$ in $X_c$ are listed from 1 to $V$
    .The intensity of each pore $\gamma_i$ is calculating by taking the
    average of the intensities of the pixel values from $X_i$ in
    the 3x3 window at the center of each pore from $X_c$
    .$X_{d_{min}}$ is an initial copy of $X_c$
 1: for $i = 1$ to $V$ from $X_c$ do
 2:   if ($i$ in $X_{d_{min}}$) then
 3:     Find the nearest neighbor pore $j$ in $X_{d_{min}}$, where
        $j \neq i$, with the smallest Euclidean distance $E_D(i, j)$
        using $P_i$ and $P_j$
 4:     if ($E_D(i, j) < d_{min}$) then
 5:       if ($\gamma_i > \gamma_j$) then
 6:         Remove $j$ from $X_{d_{min}}$
 7:         Return to step 3
 8:       else if ($\gamma_i \leq \gamma_j$) then
 9:         Remove $i$ from $X_{d_{min}}$
10:       end if
11:     end if
12:   end if
13: end for
14: return $X_{d_{min}}$
```

B. Procedure for Identifying and Removing the False Pores from $X_{d_{min}}$ Based on the Criterion of Local Mean and Standard Deviation

In this subsection, we develop a procedure for identifying as many false pores as possible in the pore map $\mathbf{X}_{d_{min}}$ based on the local means and standard deviations of the pores corresponding to the centroids still remaining in the map $\mathbf{X}_{d_{min}}$ after carrying out the procedure of subsection 3.3.2.A. Generally, the intensity of a false pore is lower than that of a true pore in a given neighborhood of the fingerprint image. The standard deviation of the pore intensities is an indication of the degree to which the intensity of a true pore could be lower than the mean intensity in the neighborhood. In our method for removing a false pore, we require that its intensity must be less than a threshold value given by $t = \text{ɱ} - c\text{б}$, where ɱ and б are the mean and standard deviation of the pore intensities within a given neighborhood of the fingerprint image and c is a parameter that needs to be determined suitably. We determine the value of c empirically for the entire dataset. The images of the dataset are divided into $W \times W$ windows. First, for a given image in the dataset, we determine a value of c so that the accuracy that all the pores removed from that image are indeed false pores is more than 96%. Then, a value c_{avg}=1.12 that is the average of the values of c so determined for the individual images is used to remove pores from each of the windows of a test image. Removing all the false pores from $\mathbf{X}_{d_{min}}$ using this procedure results in the final pore map $\mathbf{X}_{t}$.

3.3 Experimental Results of the Proposed Pore Detection Module

In this section, first, the dataset used and the procedure for the training of the proposed network along with the software and hardware platforms and the figures of merit for the evaluation procedure of the proposed and other schemes considered are described. Then, the results of the various experiments performed using the proposed pore detection scheme are presented and analyzed, and its performance is compared with that of the existing state-of-the-art schemes for pore detection.

3.3.1 Training Details and Evaluation Procedure for the Proposed Pore Detection

In our experiments, images from the PolyU High-Resolution-Fingerprint (HRF) database [113], are used for the training and testing of the proposed network and for the comparison of its performance with that of the other networks. The images in this database are acquired using an optical touch-based sensor device. The database has 30 fingerprint images of size 240 × 320 pixels with a resolution of 1200 dpi. The database also has a ground truth set containing the coordinates of the central position of each pore of the fingerprint images, as perceived by human experts. This ground truth set contains a total of 12767 coordinates of the pores in all the fingerprint images in the database.

The fingerprint images are divided into training and testing sets using k-fold cross-validation [114] with k = 5. Therefore, in our experiments, we have five different training and testing sets. We make sure that none of the 6 fingerprint images in each of the 5 testing sets are repeated and that they are different from those in other testing sets. Thus, we have five different training models of the proposed network each of which is evaluated using a unique testing set. The results of the five evaluations are then averaged to obtain the performance of the network.

In the proposed network, we generate a labeled pore intensity map corresponding to a ground truth fingerprint image by setting all the pixel values in the 3×3 window centered at the coordinates of each true pore of the ground truth image to 1 and setting all the remaining pixels in the ground truth image to a value of 0. The fingerprint images in the training set as well as in their corresponding labels are partitioned into $p \times p$ patches using a random patch extraction with overlapping method [115], which ensures that each patch stays inside the image and all the patches together in an image cover the entire image. Therefore, in this method, regardless of the value of p, there does not exist a boundary problem. A set of 142,800 patches is obtained by randomly extracting 5,950 patches in each of the training fingerprint images. Some of the patches of a fingerprint image may overlap, i.e., the patches may contain the same part of a fingerprint image. During the testing phase, however, each test image is partitioned into non-overlapping patches of size $p \times p$. In this case, depending on the value p, there may exist a situation in which a patch may not lie totally inside the image in order for all the pixels of the image to be covered by one or more patches. If in the formation of a patch on the boundary, there are not a sufficient number of pixels within the image, then the boundary pixels are simply repeated to complete the formation of that patch. It should be pointed out that the same value of p is used both in the training and testing phases of the network. In the testing phase, the network outputs a pore map corresponding to a single patch of the given input test image. When the pore maps of all the patches of a test image have been obtained, they are placed in a non-overlapping manner to compose a pore intensity map corresponding to that test input image. Data augmentations [116] of the patches through their rotations by 90° and 180°, flipping and Gaussian blurring are utilized to enrich the training dataset and boost the network performance. The convolution kernels with spatial support of $s \times s$ are randomly initialized with a Gaussian distribution having a zero mean and a variance of $\sqrt{\frac{2}{s^2 n}}$, where n is the layer width. A binary cross-entropy loss function is optimized to train the parameters of the network. The binary cross-entropy loss function given by [117]

$$L = -\frac{1}{N_B}\sum_{i=1}^{N_B}(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \tag{3.6}$$

where y_i represents the i$^{\text{th}}$ labeled patch in a batch of size N_B and $\hat{y}_i$ represents the corresponding predicted patch, is used to train the network. We employ the stochastic gradient descent (SGD) algorithm with the Nesterov acceleration scheme [118] to update the parameters and minimize the loss. In our training, the momentum parameter and initial learning rate are set as 0.9 and 0.1, respectively, and the learning rate is decreased by 0.01 after every 10 epochs until the rate becomes 0.01, beyond which it is not decreased farther. The network is trained with a batch size of 34 and a maximum number of 100 epochs. The facts that the proposed network is not very deep, it contains a modest number of parameters, the high-resolution training images help the training process [119], [120], [121], and that each training image is undergone through different kinds of augmentations, a training set constructed using the 24 fingerprint images is sufficient to adequately train the proposed pore detection network. The experiments are performed in a Python environment on a supercomputer with 2.2 GHz Intel E5-2650 v4 Broadwell CPU, 125 GB RAM and NVIDIA P100 Pascal (12GB HBM2 memory) GPU.

The figures of merit that are used for the performance evaluation are as follows:

1. True detection rate (R_T), which represents the ratio of the number of the true pores detected to the number of actual true pores present in the ground truth of a fingerprint image [60].
2. False detection rate (R_F) indicates the ratio of the number of false pores detected to the total number of detected pores [60].
3. The number of the parameters used in the CNN and the number of FLOPs, which indicate the computational cost of the CNN network.

The optimal values for R_T and R_F are one and zero, respectively. A high true detection rate and low false detection rate are indicative of a superior performance of the pore detection system, and small standard deviations of R_T and R_F represent a good robustness of the system.

3.3.2 Performance Evaluation

The objective of this subsection is to study the performance of the proposed pore detection scheme and compare it with that of other schemes in the literature. We conduct this study in three parts. In the first part, we study the impact of the patch size of the images and the depth of the network on the performance of the proposed scheme. In the second part, we study the influence of replacing either the proposed network or the proposed postprocessing method with the one used in other neural network-based schemes. In the third part, we compare both the quantitative and qualitative performance of the proposed scheme with those of the other schemes that are neural network-based or otherwise. The computational times of the network part and the postprocessing part of the proposed scheme along with its network complexity are also compared with those of the other network-based schemes.

In the first part of the performance evaluation, we first perform experiments on the proposed scheme by using different patch sizes of the input images in the dataset as input to the proposed network. Once the pore intensity maps for all the test images have been obtained, we move on to the second part of the proposed scheme that uses the hyperparameters N (the window size used in the binarization step) and W (the window size used in the refinement step). In this part of the scheme, for a test image, the pore intensity map obtained from the network patches trained by using the patches of a given size p is then used to obtain $\mathbf{X}_t$ using a given value of the parameter N and a given value of the parameter W. Using this final map of pore centroids, the values of R_T and R_F are obtained for a given test image. These values of the metrics are averaged over 30 test images. We have performed the experiments on the proposed scheme using different values for the patch size p (10, 20, 30, 40, 50, 60, 70, 80, 90, and 100) in Part 1 of the scheme and for each value of p with various values of the hyperparameters N (20, 30, 40, 50, and 60) and W (40, 50, 60, 70, and 80) in Part 2. Table 3.1 lists the average values of R_T and R_F and their standard deviations for each of the patch sizes used for the network training and testing, but only for those particular values of N and W that provide the best performance in terms of R_T and R_F for that patch size. From this table, it is seen that the proposed scheme of pore detection provides the best performance in terms of R_T and R_F for p =40, N =40 and W =60. Table 3.1 also provides the network complexity in terms of the number of floating-point multiplication-addition operations (FLOPs). It is worth noting that the increase in the number of FLOPs by using a patch size beyond 40 × 40 actually results in deteriorating the performance of the scheme. Therefore, 40 × 40 is the optimum patch size for the best performance of the proposed scheme.

Next, we perform an experiment on the proposed network in which its depth is increased by adding additional feature extraction stages. The results are shown in Table 3.2. It is seen from this table that as the number of feature extraction stages in the network increases beyond three stages, the improvement in its performance is only minimal. However, its complexity in terms of the number of FLOPs increases very significantly. Specifically, the number of FLOPs increases by more than 4 and 11 times when the number of feature extraction stages is increased to 4 and 5, respectively. On the other hand, if the number of feature extraction stages is reduced to two, the performance of the network is severely affected. Thus, the network provides optimum performance with three feature extraction stages.

In the second part of the performance evaluation, we examine the impact of the proposed network and the proposed postprocessing method individually on the performance of the proposed pore detection scheme, as well as on the performances of the neural network-based schemes of [60], [61] and [62]. Table 3.3 provides the results of this study in terms of the performance metrics R_T and R_F. The results in this table are divided into four blocks corresponding to the proposed and the schemes of [60], [61] and [62]. In each of the blocks 2, 3 and 4, the performance results of the corresponding scheme along with that of the same scheme in which either its network or its postprocessing method is replaced by the one proposed in our scheme. By comparing the first lines in each of the blocks of this table, it is observed that the proposed scheme gives the best performance among all the neural network-based schemes. By comparing the results of the first and second lines of the blocks 2, 3 and 4 of this table, it is seen that each of the schemes of [60], [61] and [62] can significantly benefit if the networks of these schemes are replaced by the proposed network. Similarly, by comparing the results of the first and third lines of the blocks 2, 3 and 4, it is seen that each of the schemes of [60], [61] and [62] can improve the performance if its postprocessing method is replaced by the proposed postprocessing method. In summary, the study of this experimental part clearly shows that both the proposed network and the postprocessing method have a significant impact on the performance of the proposed pore detection scheme.

Table 3.1: Performance of the proposed scheme with different patch sizes of the images input to the network.

Patch size	10 × 10	20 × 20	30 × 30	40 × 40	50 × 50	60 × 60	70 × 70	80 × 80	90 × 90	100 × 100
Hyperparameters	$N = 50$	$N = 40$	$N = 50$	$N = 40$	$N = 60$	$N = 60$	$N = 40$	$N = 40$	$N = 40$	$N = 60$
	$W = 60$	$W = 60$	$W = 70$	$W = 60$	$W = 80$	$W = 80$	$W = 60$	$W = 80$	$W = 60$	$W = 70$
R_T	91.84	93.16	95.97	**96.69**	96.51	96.43	95.93	95.51	94.22	93.96
(σ)	(1.32)	(1.48)	(1.49)	**(1.52)**	(1.83)	(1.99)	(2.57)	(2.74)	(3.2)	(3.51)
R_F	10.02	6.55	4.15	**4.18**	4.2	4.68	5.33	5.87	6.05	7.04
(σ)	(3.22)	(1.97)	(1.41)	**(1.44)**	(1.9)	(1.83)	(2.41)	(2.6)	(3.95)	(4.02)
Number of FLOPs	0.495M	0.99M	1.485M	***1.98M***	2.475M	2.97M	3.465M	3.96M	4.455M	4.95M

Table 3.2: Performance of the proposed scheme with various network depths.

Number of Feature extraction stages	R_T (σ)	R_F (σ)	Number of FLOPs
2	91.68 (4.93)	6.62 (5.10)	0.86M
3	**96.69 (1.52)**	**4.18 (1.44)**	**1.98M**
4	96.89 (1.5)	4.03 (1.41)	9.49M
5	96.95 (1.49)	3.95 (1.40)	22.15M

Table 3.3: Comparison of performances of the proposed and other neural network-based schemes.

Scheme	Detail of the scheme	R_T (σ)	R_F (σ)
Proposed scheme	Proposed network + proposed postprocessing	**96.69 (1.52)**	**4.18 (1.44)**
Scheme [60]	Network of [60] + postprocessing of [60]	84.69 (7.81)	15.31 (6.2)
Modified scheme [60] – 1	Proposed network + postprocessing of [60]	93.19 (3.04)	7.88 (3.55)
Modified scheme [60] - 2	Network of [60] + proposed postprocessing	88.21 (6.03)	5.46 (2.74)
Scheme [61]	Network of [61] + postprocessing of [61]	93.09 (4.63)	8.64 (4.15)
Modified scheme [61] – 1	Proposed network + postprocessing of [61]	95.47 (2.58)	6.52 (3.56)
Modified scheme [61] – 2	Network of [61] + proposed postprocessing	94.81 (3.05)	4.53 (2.07)
Scheme [62]	Network of [62] + postprocessing of [62]	94.49 (5.41)	8.5 (4.4)
Modified scheme [62] – 1	Proposed network + postprocessing of [62]	95.89 (2.42)	6.42 (3.41)
Modified scheme [62] - 2	Network of [62] + proposed postprocessing	95.22 (3.15)	4.38 (2.10)

In the third part of our experimental study, we compare the quantitative and qualitative performance of the proposed pore detection scheme with that of both the traditional schemes and neural network-based schemes. The traditional schemes that are used for comparison are those reported in [38] and [49], whereas the neural network-based schemes are the same as those used in Table 3.3, i.e., the schemes of [60], [61] and [62]. Table 3.4 gives the pore detection accuracy in terms of R_T and R_F provided by the various schemes. It is seen from this table that, as in the case of other neural network-based schemes with the exception of that reported in [60], the proposed scheme outperforms the traditional schemes by significant margins. It is to be pointed out that, since the network in [60] relies only on the network width to extract the pore features, it is not able to learn the high-level features, and hence, this network is not able to provide much improvement over the non-neural network-based methods. As already noted in Table 3.3, the proposed scheme provides performance much superior to that provided by the other neural network-based schemes. Table 3.4 provides the performance results using both with and without the pore refinement process introduced in Section 3.3. Note that the pore refinement process was intended to reduce the number of false pores. It is seen from the performance results given in

Table 3.4 that this objective is achieved by providing the value of R_F that is almost one-half of that provided without using the pore refinement process while maintaining almost the same value for R_T.

Table 3.4: Performance comparison against state-of-the-art methods.

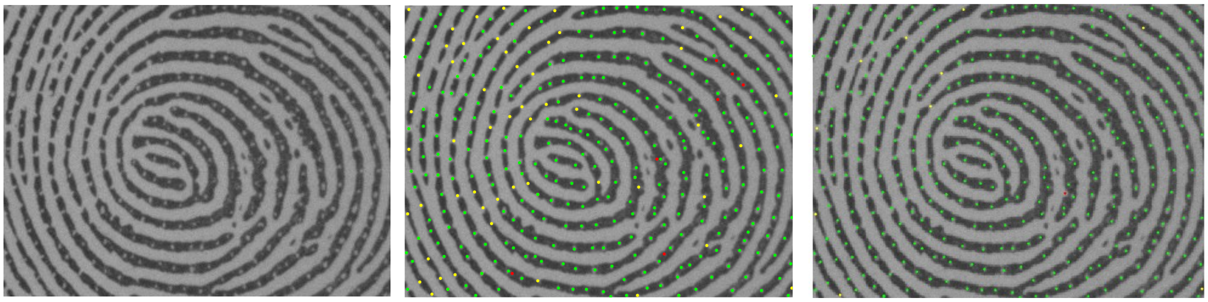
Scheme	R_T (σ)	R_F (σ)
Gabor Filter [38]	75.90 (7.5)	23.00 (8.2)
DoG [49]	80.80 (6.5)	22.20 (9.0)
DAPM [49]	84.80 (4.5)	17.60 (6.3)
Labati *et al.* [60]	84.69 (7.81)	15.31 (6.2)
DeepPore [61]	93.09 (4.63)	8.64 (4.15)
DeepResPore [62]	94.49 (5.41)	8.5 (4.4)
Proposed without Refinement	**96.85 (1.53)**	**8.69 (2.12)**
Proposed with Refinement	**96.69 (1.52)**	**4.18 (1.44)**

Table 3.5 demonstrates the average run times per image of the various CNN-based pore detection schemes for Parts 1 and 2 of the schemes individually as well as together. All the schemes have been implemented in the same software and hardware environment of the proposed scheme as stated in Section 3.4.1. It is seen from this table that the time complexities of the second part of all four schemes are about the same. On the other hand, the proposed scheme takes significantly lower time for its first part in comparison to the times taken by the other three schemes. This table also provides information on the complexity of the various neural network-based schemes in terms of the number of parameters and the number of FLOPs. Note that the number of parameters and the number of FLOPs refers only to the numbers used by the network parts of these schemes. It is particularly important to note that the superiority in performance of the proposed scheme is achieved along with its very significantly lower complexity. Specifically, the number of parameters and the number of FLOPs of the network in the proposed scheme are several orders of magnitude lower than those of the network in the scheme of [62], the best performing scheme in the literature.

Table 3.5: Comparison of average time per image and network complexity of the proposed and other neural network-based schemes.

Scheme parts	T_{CNN} (ms) Part 1: Neural Network	T_P (ms) Part 2: Postprocessing	T_T (ms)	Number of Network Parameters	Number of FLOPs used by Network
Labati *et al.* [60]	2.76	15.54	18.3	1,531	50.03M
DeepPore [61]	29.85	16.1	46.36	297,361	335.12 M
DeepResPore [62]	218.02	16.24	234.67	11M	70,438.1M
Proposed scheme	**1.08**	**15.92**	**17**	**1,288**	**1.98M**

Fig. 3.8 shows the qualitative performance of the proposed scheme and the schemes of [61] and [62] by providing visual illustrations of the pores detected. Fig. 3.8 (a) shows an original fingerprint image from the HRF dataset, whereas Figs. 3.8 (b), (c) and (d) show the results of the pore detection by the three schemes. In this figure, the pores shown in green, red and yellow colors represent, respectively, true positive, false positive and false negative pores. It is seen from this figure that the proposed scheme is more successful in detecting the true pores by providing only one false positive pore and much fewer false negative pores than that provided by the other two schemes.



(a) (b) (c)

Figure 3.8: Qualitative performance of the proposed pore detection scheme. (a) Original fingerprint image. The pore detection results of (b) DeepResPore (c) Proposed work. The green, yellow, and red dots in these images represent the true positive, false negative and false positive, respectively.

3.4 Summary

In recent years, advancements in pore detection have led to the emergence of hybrid approaches that incorporate convolutional neural networks (CNNs) to enhance accuracy. However, these approaches faced challenges with incomplete or distorted pores due to limitations in feature learning and post-processing. This chapter has proposed an ultralight hybrid pore detection scheme resulting in high representation with low complexity. The first part has developed a neural network architecture to generate and extract low, middle and high level hierarchical features in a residual framework using depthwise convolutional layers and then to fuse them to obtain a very rich and highly representational set of pore features leading to an accurate pore intensity map. The second part has leveraged the existing knowledge about fingerprint pores, such as the pore intensity, their variations in terms of shape and size from one region to another region of the fingerprint image and the minimum distance between neighboring pores, for an accurate centroid determination. Extensive experiments on the PolyU High-Resolution-Fingerprint database have been conducted and the results have demonstrated the performance of the proposed scheme to be superior to that provided by the traditional and state-of-the-art CNN-based schemes.

Chapter 4

Proposed Pore Feature Extraction and Matching Module

4.1 Introduction

In this chapter, we propose a scheme for pore feature extraction and matching [122], comprising three segments, namely, pore feature extraction and representation, pore matching and pore matching refinement. The proposed module aims at an optimal performance with minimal complexity by focusing primarily on the first two segments. The first segment develops a low-complexity residual block that is embedded in a deep convolutional neural network, designed to yield highly representative pore features. The network makes a judicious use of depthwise and, depthwise separable convolution operations, and Max-pooling so as to curtail the number of parameters and arithmetic operations. The second segment introduces a novel metric, devised to measure the pore similarity by merging angles, magnitude differences, and Euclidean distances into a single metric. This new metric is then applied on the pore representation vectors obtained from the first segment to produce the pore correspondence list of the matches of the pores in the query and template fingerprint images.

4.2 Proposed scheme

A proposed pore feature extraction and matching module aims at providing a decision for the query image to have matched (i.e., a value of 1) or not to have matched (i.e., a value of 0) with the template fingerprint image in question. The pore feature extraction and matching module is divided into three segments. In the first segment (Segment 1), given the query fingerprint image and the entire set of *N* pores detected by Module 1 as input, this segment outputs a representation $\boldsymbol{PF_{Qi}}$ ($i = 1,...,N$) of each of the pores in the query image in terms of the structure of the pore as well as of its neighborhood by extracting suitable features from a patch corresponding to the pore in question using a CNN automatic feature extraction technique. The second segment (Segment 2) of the pore feature extraction and matching module is a pore matching segment, in which the feature representation $\boldsymbol{PF_{Qi}}$ of each of the pores in the query image in the entire set $\boldsymbol{PF_Q}$ and that of all the pores, but one at a time, in a single template fingerprint image in the set $\boldsymbol{PF_T}$ from the database are matched based on some criterion of their similarities to establish a correspondence between the pores of each pair formed by taking one pore from $\boldsymbol{PF_Q}$ and other one from $\boldsymbol{PF_T}$. The output of the module of this segment is a set C of *c_i's*, where each c_i (i = 1, .., K) corresponds to a pair of matched pores with the first one in the pair representing the x-y coordinates of a pore in the query fingerprint image and the second one representing the x-y coordinates of the matched pore in the template fingerprint image. The third segment (Segment 3) of the pore feature extraction and matching module is a pore matching refinement and fingerprint recognition segment. In this segment, a refined set $\boldsymbol{RC}$ of the correspondence between the pores of the query image and those that are matched in the template image is obtained by removing those pairs in which the pores are falsely matched in Segment 2. Finally in this segment, the query fingerprint image is recognized based on the refined set $\boldsymbol{RC}$. If this set contains one or more pairs of matched pores, i.e., $L \geq 1$, the query image is considered to be matched to the template image, otherwise (i.e., L=0), the two fingerprint images in question are considered as not matched. Fig. 4.1 shows an example (Illustration 1) of the pores from a query image and a template image that have been matched by the operation of Segment 2. In this illustration of the images, the pairs of pores assumed to have been matched after Segment 2 are joined by the red and yellow lines. In Illustration 2 of Fig. 4.1, the same two fingerprint images have been shown with the pairs of pores retained as matched pairs after the operation of Segment 3 (i.e., after the pore matching refinement operation) joined by the yellow lines. We next develop each of these three segments of the pore feature extraction and matching module with reference to Fig. 4.1.

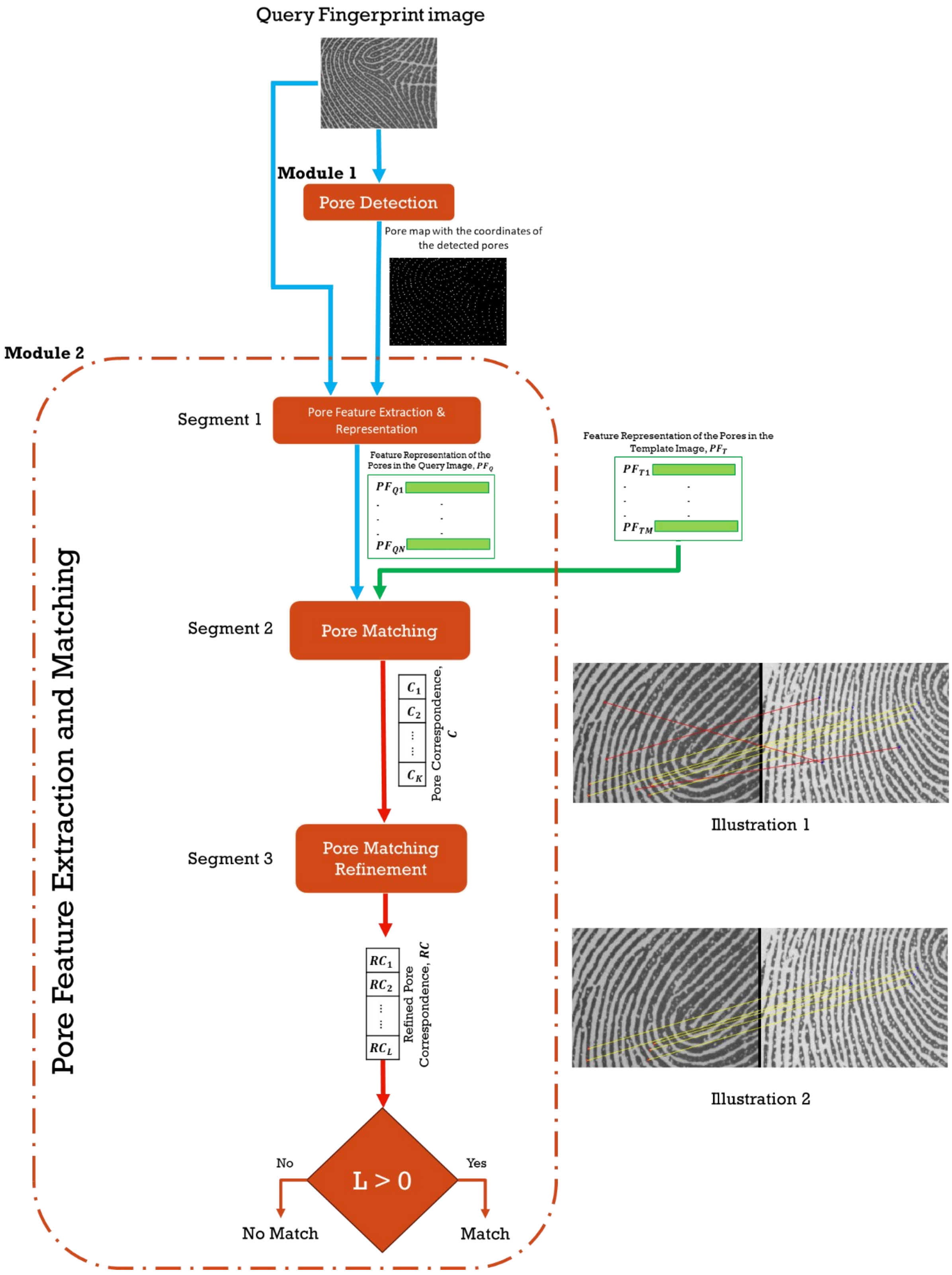

Query Fingerprint image
Module 1
Pore Detection
Pore map with the coordinates of the detected pores
Module 2
Pore Feature Extraction and Matching
Segment 1
Pore Feature Extraction & Representation
Feature Representation of the Pores in the Query Image, PF_Q
PF_{Q1}
PF_{QN}
Feature Representation of the Pores in the Template Image, PF_T
PF_{T1}
PF_{TM}
Segment 2
Pore Matching
C_1
C_2
C_K
Pore Correspondence, C
Segment 3
Pore Matching Refinement
RC_1
RC_2
RC_L
Refined Pore Correspondence, RC
L > 0
No
Yes
No Match
Match
Illustration 1
Illustration 2

Figure 4.1: General module of pore-based fingerprint recognition highlighting the pore feature extraction and matching module.

4.2.1 Segment 1: Feature Extraction and Representation of Pores Using a Convolutional Neural Network

In this section, a novel CNN model for the extraction and representation of fingerprint pores is proposed. The main objective in the design of the CNN model is to ensure robustness in capturing a discriminative set of features for the fingerprint pores and to generate a unique representation for each of them in a computationally efficient manner. The model design involves several key ideas specifically relevant to the task of fingerprint recognition using pores.

1. Hierarchical Feature Extraction: The network is designed with multiple stages that aim to produce diverse levels of hierarchical pore features. This approach enables the extraction of the features of the pores at different scales and levels of detail.

2. Concatenative Fusion: A concatenative fusion of the feature maps at various levels and scales is employed to provide a compact and highly discriminative representation of the features of a pore.

3. Depthwise or Depthwise Separable Convolutions: In view of the fact that pore features are local features, depthwise or depthwise separable convolutions are employed, since it can extract local features more efficiently in comparison to the standard convolution, and in addition, they do so at a lower computational cost.

4. Identification-Verification Loss Function: An identification-verification loss function [123] is utilized to generate the pore feature representation. The identification-verification loss function serves two main purposes:

   (a) Identification: The identification task involves classifying an input fingerprint pore patch image into one of the numerous individual identity classes. This task is a multi-class classification in which the model is required to accurately assign a correct identity label to the input pore patch from a set of pore patches belonging to the various classes.

   (b) Verification: The verification task involves classifying the two patches in a pair as either belonging to the same class or two different classes, which is essentially a binary classification problem.

The main advantages of using this identification-verification loss function for the training of the features extraction and representation network are as follows:

(i) This loss function promotes the learning of highly discriminative features specific to fingerprint pore identities. By jointly optimizing for identification and verification, the model learns to extract features that capture both inter-class variations (for identification) and intra-class similarities (for verification). This helps to generate a compact and powerful feature representation for fingerprint pore.

(ii) This loss function is able to minimize the distance and orientation between feature representations of the pores belonging to the same class and maximize these quantities when the pores belong to different classes. This results in a trained model that has a better capability of distinguishing the various pore classes and hence is able to produce more accurate pore feature representations.

5. Deep Supervision: Deep supervision involves incorporating additional supervision signals from the intermediate stages of a network, in addition to the final output signal of the network [124]. A trained network for generating pore feature representations must be capable of capturing local and global details of the pores. Local details encompass attributes such as pore size, shape, orientation, and specific ridge patterns surrounding the pore, while global details involve factors such as the distribution of pores within the fingerprint, overall pore density in specific regions, or arrangement of pores along ridge structures. By leveraging deep supervision, a network can more effectively and accurately capture the local and global features of the fingerprint pores, resulting in a more comprehensive and discriminative pore feature representations.

The proposed network model is shown in Fig. 4.2. The input to the network is a pore patch $\boldsymbol{Y_I}$ of size 31×31 cropped around a pore of the fingerprint image and the objective of the network is to produce at the output a feature representation vector $\boldsymbol{Y_O}$ of size 256 representing a unique identity for the pore in question in the patch. The network consists of five stages, producing features at increasingly higher levels of hierarchy using an increasingly larger number of convolution filters. We fuse the features in each stage in a residual framework to learn the local features of the pore and fuse the features obtained from the various hierarchical levels concatenatively to learn global features. The fusion of the features both locally and globally and using them in a deep supervision framework is aimed at extracting highly discriminating features, and thus, providing a unique feature representation for the pore.

In order to produce a diverse range of features from the input patch $\boldsymbol{Y_I}$ at low computational cost, in the first stage, we employ two depthwise separable convolutional layers (D_S Conv) each using 32 filters of kernel size 3×3. The outputs of the two layers are then additively fused and then max pooled with 3×3 kernels and a stride of 2 to produce the output $\boldsymbol{Y_1}$ of the stage consisting of 32 channels each of spatial size 21×21.

In order to increase both the number and diversity of the features extracted from the first stage, in each of the stages 2 to 5, we employ two depthwise blocks (D_Block) followed by a max-pooling layer. The architecture of the depthwise block consists of two layers, a depthwise convolution layer and a depthwise separable convolutional layer. The depthwise convolutional layer of the first D_Block of each of these stages employes two filters of kernel size 3×3 for each of its input channels, whereas the second D_Block uses only one filter of the same kernel size. On the other hand, the depthwise separable convolutional layer in both the D_Blocks of a stage uses only one 3×3 filter for each of its input channels. The outputs of the depthwise and depthwise separable convolutional layers of a D_Block are additively fused to produce the output of the D_block. Thus, the number of channels in the output of a stage is twice the number of channels in the input of that stage.

In a typical fingerprint image, the variations in pores are more prominent on a global scale rather than on a local level. Thus, selecting a smaller patch size would yield feature representations that are less distinctive. On the other hand, choosing a very large pore patch size would result in a large computational complexity. Therefore, for our proposed architecture, we choose the size of the input patches to be 31×31. The use of a large number of stages in our architecture can facilitate the extraction of high-level features. For our architecture, we use the largest possible number of stages, which is constrained to be 5, in view of the fact that the size of the input patch is 31×31 and there is a pooling layer in each of the stages, thus leading to a size of 1×1 for each of the channels in the output Y_5 of the last stage. In order for our proposed network to produce a very rich set of features, the features produced by the various stages are fused, which is done by a process of global max-pooling (i.e., selecting the maximum value) each of the channels of the outputs, $\boldsymbol{Y_1}$, $\boldsymbol{Y_2}$, $\boldsymbol{Y_3}$ and $\boldsymbol{Y_4}$, of the stages 1, 2, 3 and 4 and concatenating the resulting outputs $\boldsymbol{Y_{1g}}$, $\boldsymbol{Y_{2g}}$, $\boldsymbol{Y_{3g}}$ and $\boldsymbol{Y_{4g}}$ with $\boldsymbol{Y_5}$. Thus, the output of the concatenation is $\boldsymbol{Y}_C = (\boldsymbol{Y_{1g}}\ \boldsymbol{Y_{2g}}\ \boldsymbol{Y_{3g}}\ \boldsymbol{Y_{4g}}\ \boldsymbol{Y_5})$ with a total of 992 channels each of size 1x1. The final step is to produce from $\boldsymbol{Y}_C$ an output vector $\boldsymbol{Y}_O$ representing the input pore, using a fully connected neural network (FC_Final) of 256 neurons. Therefore, $\boldsymbol{Y}_O$ is a 256-dimensional vector representing the pore in question in the input pore patch $\boldsymbol{Y}_I$. The details of the architecture in terms of the kernel size, the number of filters and the number of parameters used in the various layers are provided in Table 4.1.

In each of the abovementioned stages, a non-linear activation function (ReLU) is applied after each convolutional layer and in order to normalize the distribution of the features produced by a convolutional layer, a batch normalization (BN) is applied. The residual framework used in these stages curtails the gradient vanishing problem in the proposed deep network leading to better training of the network parameters, and hence, the generation of more representational features. As the features produced at all the hierarchical levels are important, the proposed network employes deep supervision for its training. Therefore, in order to carry out the deep supervision training of the network, we need to produce 256-dimensional representations $\boldsymbol{Y}_{FC1}, \boldsymbol{Y}_{FC2}, \boldsymbol{Y}_{FC3}, \boldsymbol{Y}_{FC4}$ and $\boldsymbol{Y}_{FC5}$ from the outputs $\boldsymbol{Y}_1, Y_2, Y_3, Y_4\ and\ Y_5$, respectively, of the five stages using fully connected neural networks each using 256 neurons. It is to be noted that the five fully connected neural networks are needed only for the training of the network.

Table 4.1: Architecture detail of the proposed pore matching network.

Size of input pore patch Y_I: 31 × 31				
	Layer name	Kernel size and no. filters	Size and the no. output channel	No. Parameters
Stage 1	First D_S Conv	3 × 3, 32	31 × 31, 32	320
	Second D_S Conv	3 × 3, 32	31 × 31, 32	320
	Max-Pooling	3 × 3, stride 2	Y_1: 21 × 21, 32	0
Stage 2	First D_Block:			
	D Conv	3 × 3, 64	21 × 21, 64	576
	D_S Conv	3 × 3, 64	21 × 21, 64	640
	Second D_Block:			
	D Conv	3 × 3, 64	21 × 21, 64	576
	D_S Conv	3 × 3, 64	21 × 21, 64	640
	Max-Pooling	3 × 3, stride 2	Y_2: 11 × 11, 64	0
Stage 3	First D_Block:			
	D Conv	3 × 3, 128	11 × 11, 128	1152
	D_S Conv	3 × 3, 128	11 × 11, 128	1253
	Second D_Block:			
	D Conv	3 × 3, 128	11 × 11, 128	1152
	D_S Conv	3 × 3, 128	11 × 11, 128	1253
	Max-Pooling	3 × 3, stride 2	Y_3: 6 × 6, 128	0
Stage 4	First D_Block:			
	D Conv	3 × 3, 256	6 × 6, 256	2304
	D_S Conv	3 × 3, 256	6 × 6, 256	2560
	Second D_Block:			
	D Conv	3 × 3, 256	6 × 6, 256	2304
	D_S Conv	3 × 3, 256	6 × 6, 256	2560
	Max-Pooling	3 × 3, stride 2	Y_4: 3 × 3, 256	0
Stage 5	First D_Block:			
	D Conv	3 × 3, 512	3 × 3, 512	4608
	D_S Conv	3 × 3, 512	3 × 3, 512	5120
	Second D_Block:			
	D Conv	3 × 3, 512	3 × 3, 512	4608
	D_S Conv	3 × 3, 512	3 × 3, 512	5120
	Max-Pooling	3 × 3, stride 2	Y_5: 1 × 1, 512	0
Global Pooling	GP1	21 × 21	Y_{1g}: 1 × 1, 32	0
	GP2	11 × 11	Y_{2g}: 1 × 1, 64	0
	GP3	6 × 6	Y_{3g}: 1 × 1, 128	0
	GP4	3 × 3	Y_{4g}: 1 × 1, 256	0
Concatenation	C		Y_C: 1 × 1, 992	0
Fully Connected layer	FC-1	256-D	Y_{FC1}: 256	3612672
	FC-2	256-D	Y_{FC2}: 256	1982464
	FC-3	256-D	Y_{FC3}: 256	1179648
	FC-4	256-D	Y_{FC4}: 256	589824
	FC-5	256-D	Y_{FC5}: 256	131072
	FC-Final	256-D	Y_O: 256	253952

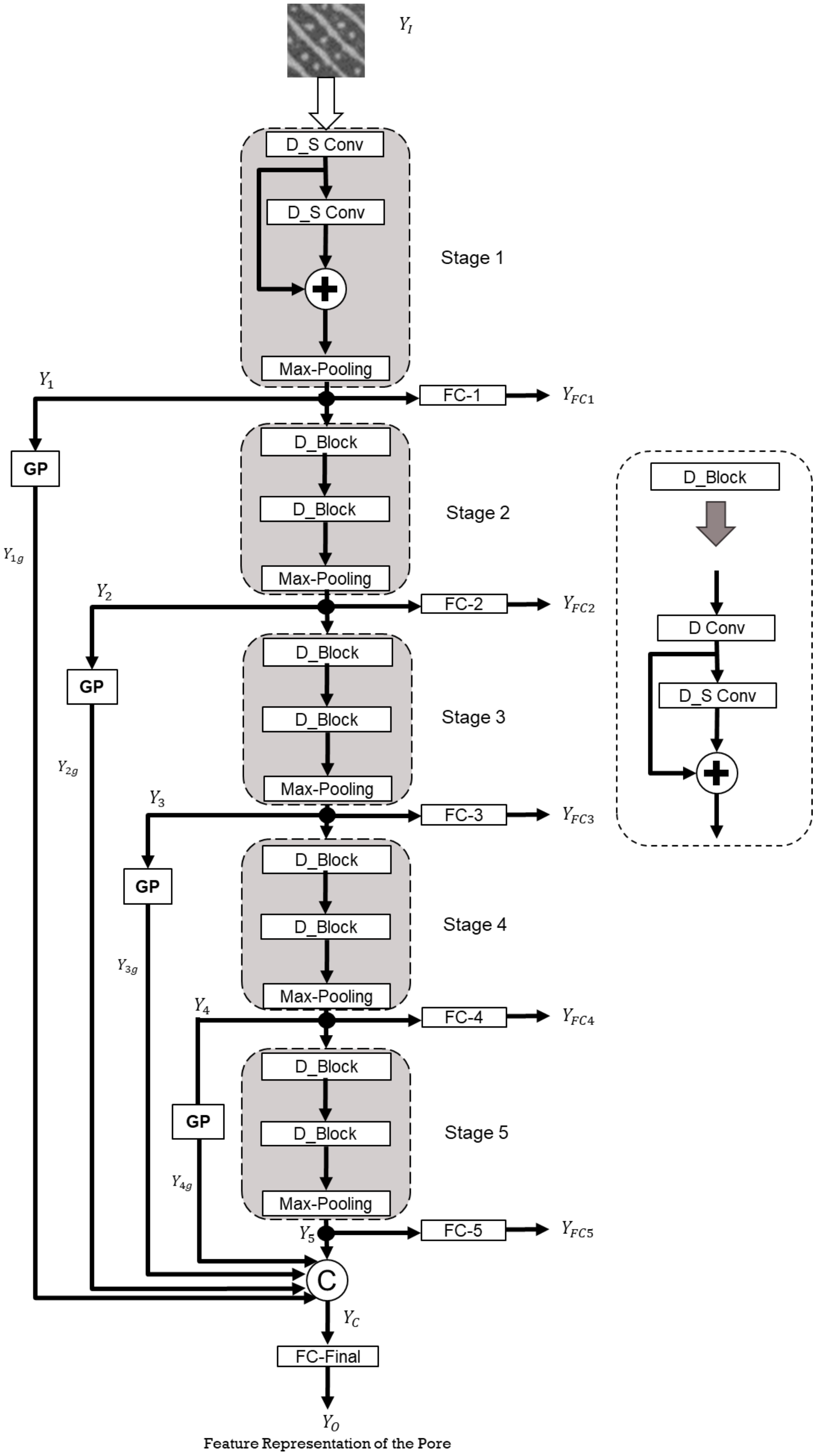


Figure 4.2: Proposed network architecture for pore feature extraction and representation.

4.2.2 Segment 2: Pore Matching

At this point, we have pore representations of all the pores in the query image and those of all the pores in a template image resulting from Segment 1 of Module 2. In this section, we present a technique for matching a pore from the query image with a pore from the template image using their feature representations.

The most common method of matching two entities represented by N-dimensional vectors, $\boldsymbol{A}$ and $\boldsymbol{B}$, is to compute the Euclidean distance (ED) between them:

$$ED(\boldsymbol{A},\boldsymbol{B}) = \sqrt{\sum_{n=1}^{N}(\boldsymbol{A}_n - \boldsymbol{B}_n)^2} \tag{4.1}$$

and use it to determine the similarity between them. An N-dimensional vector represents a point in an N-dimensional space. Fig. 4.3 (a) shows an example of the Euclidean distance matric between two 2-dimensional vectors $\boldsymbol{A}$ and $\boldsymbol{B}$. The highest similarity between the two vectors occurs when both of them are represented by the same point in N-dimensional space, meaning that the value of ED is 0. However, ED, as defined by equation (4.1), is not a very effective metric to determine the degree of similarity between two vectors, since this metric ignores the orientation of the vectors. To illustrate this point, we consider an example in 2-dimensional space, shown in Fig. 4.3 (b), in which we are interested in determining the degrees of similarities of two vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ with the vector $\boldsymbol{M}$. It is seen from this figure that $ED(\boldsymbol{M},\boldsymbol{P})$= $ED(\boldsymbol{M},\boldsymbol{Q}) = 3$, meaning that both $\boldsymbol{P}$ and $\boldsymbol{Q}$ are equally similar to the vector $\boldsymbol{M}$.

Another metric, known as triangle similarity metric [125], that has been used to measure the similarity between the two vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ is the area of the triangle formed by these two vectors.

$$TS(\boldsymbol{A},\boldsymbol{B}) = \frac{|\boldsymbol{A}||\boldsymbol{B}|\sin(\theta)}{2} \tag{4.2}$$

where $|\boldsymbol{A}| = \sqrt{\sum_{n=1}^{N}\boldsymbol{A}_n^2}$, $|\boldsymbol{B}| = \sqrt{\sum_{n=1}^{N}\boldsymbol{B}_n^2}$, an $\theta = cos^{-1}\left(\frac{\sum_{n=1}^{N}\boldsymbol{A}_n.\boldsymbol{B}_n}{|\boldsymbol{A}||\boldsymbol{B}|}\right)$d. Fig. 4.3 (a) also shows the area between vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ representing the value of the TS metric. As with the Euclidean distance, TS metric also approaches zero as the two vectors approach each other, meaning that the two vectors become increasingly more similar, and the highest similarity is achieved when the two vectors overlap. However, the TS metric has the same problem as ED

metric, in that TS metric may provide the same value for the similarity of two distinct vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ with respect to $\boldsymbol{M}$. Fig. 4.3 (c) shows an example of such a situation. It is seen from this figure that even though $\boldsymbol{P}$ and $\boldsymbol{Q}$ are distinct vectors, $TS(\boldsymbol{M},\boldsymbol{P})$= $TS(\boldsymbol{M},\boldsymbol{Q})$ = 1, meaning that both vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ are equally similar to the vector $\boldsymbol{M}$ in term of triangle similarity.

However, it is seen from the example of Fig. 4.3 (b) that even though $ED(\boldsymbol{M},\boldsymbol{P})$= $ED(\boldsymbol{M},\boldsymbol{Q})$ = 3, $TS(\boldsymbol{M},\boldsymbol{P})$ = 2.4 and $TS(\boldsymbol{M},\boldsymbol{Q})$ = 4.85. Similarly, from Fig. 4.3 (c), it is seen that even though $TS(\boldsymbol{M},\boldsymbol{P})$= $TS(\boldsymbol{M},\boldsymbol{Q})$ = 1, $ED(\boldsymbol{M},\boldsymbol{P})$ = 2.96 and $ED(\boldsymbol{M},\boldsymbol{Q})$ = 1.63. This implies that in the case of two distinct vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$, their similarities with respect to $\boldsymbol{M}$ are not the same in terms of both the ED and TS metrics simultaneously. Also, we may have a situation such as the one shown in Fig. 4.3 (d) where $\boldsymbol{P}$ is more similar to $\boldsymbol{M}$ than $\boldsymbol{Q}$ in terms of ED, whereas $\boldsymbol{Q}$ is more similar to $\boldsymbol{M}$ than $\boldsymbol{P}$ in terms of TS.

Sector area similarity (SS) is another measure of similarity between two vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ introduced in [126]. It is defined as the area of a section of a circle of radius $(ED + MD)$ with angle θ (see Fig. 4.3 (e)):

$$SS(\boldsymbol{A},\boldsymbol{B}) = \pi.\left(ED(\boldsymbol{A},\boldsymbol{B}) + \quad MD(\boldsymbol{A},\boldsymbol{B})\right)^2.\left(\frac{\theta}{360}\right) \tag{4.3}$$

where MD is the absolute value of the difference between the magnitude of vector $\boldsymbol{A}$ and that of vector $\boldsymbol{B}$, and is given by:

$$MD(\boldsymbol{A},\boldsymbol{B}) = \left|\sqrt{\sum_{n=1}^{N}\boldsymbol{A}_n^2} - \sqrt{\sum_{n=1}^{N}\boldsymbol{B}_n^2}\right| \tag{4.4}$$

Fig. 4.3 (e) shows the sector area representing the value of the SS metric. As $\boldsymbol{A}$ approaches $\boldsymbol{B}$, $ED + MD$ → 0 and θ → 0, and therefore, SS metric approaches zero. In this measure, if we have two distinct vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$, we can’t have $SS\ (\boldsymbol{P},\boldsymbol{M})\ =\ SS\ (\boldsymbol{Q},\boldsymbol{M})$ unless the vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ are symmetrically located with respect to the vector $\boldsymbol{M}$ and they have the same magnitude. However, in the case of similarity measures of ED and TS, there are other situations in which $ED\ (\boldsymbol{P},\boldsymbol{M})\ =\ ED\ (\boldsymbol{Q},\boldsymbol{M})\ or\ TS\ (\boldsymbol{P},\boldsymbol{M})\ = TS\ (\boldsymbol{Q},\boldsymbol{M})$.

Hence, in the case of using SS measure, the only deadlock situation of determining as to which of the two vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ is more similar to the vector $\boldsymbol{M}$ is only when $\boldsymbol{P}$ and $\boldsymbol{Q}$ are of equal magnitude and symmetrically located with respect to $\boldsymbol{M}$. While the TS measure emphasizes the geometric relationship formed by the magnitudes of two vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ and the angle between them, the SS measure provides a more comprehensive geometric assessment by incorporating both the Euclidean distance and the magnitude difference information between the vectors $\boldsymbol{A}$ and $\boldsymbol{B}$, in addition to the angle between them. However, SS lacks the information on the magnitudes of the two vectors individually, which is contained in the TS measure. Therefore, a suitable combination of these two measures can be expected to capture various geometric characteristics of the vectors $\boldsymbol{A}$ and $\boldsymbol{B}$, providing a more accurate similarity between them, and thus resulting in reducing the number of deadlock cases discussed above. In [126], a combination of the triangle similarity and sector similarity measures has been proposed for measuring the similarity levels among the documents. This metric is defined as

$$TS\_SS(\boldsymbol{A},\boldsymbol{B}) = TS(\boldsymbol{A},\boldsymbol{B}).SS(\boldsymbol{A},\boldsymbol{B}) \tag{4.5}$$

As an illustration of the comparison of the various similarity measures discussed in this section, we apply these measures to the vectors in Figs. 4.3 (b), (c) and (d) and determine wither or not $\boldsymbol{P}$ is more similar to $\boldsymbol{M}$ than $\boldsymbol{Q}$. The results of the comparison are given in Table 4.2. It is seen from this table that the conclusions reached by applying TS_SS measure always align with the majority of the conclusions reached by the other three similarity measures.

Table 4.2: Similarity measure to the vectors $\boldsymbol{M}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$ in Figs. 4.3 (b), (c) and (d).

Figure	ED measure	TS measure	SS measure	TS_SS measure
4.3 (b)	$\boldsymbol{P},\boldsymbol{Q}$	$\boldsymbol{P}$	$\boldsymbol{P}$	$\boldsymbol{P}$
4.3 (c)	$\boldsymbol{Q}$	$\boldsymbol{P},\boldsymbol{Q}$	$\boldsymbol{Q}$	$\boldsymbol{Q}$
4.3 (d)	$\boldsymbol{P}$	$\boldsymbol{Q}$	$\boldsymbol{P}$	$\boldsymbol{P}$

For the example shown in Fig. 4.3 (b), $TS\_SS(\boldsymbol{M},\boldsymbol{P})$ = (2.4) · (3.53) = 8.47 and $TS\_SS(\boldsymbol{M},\boldsymbol{Q})$ = (4.85) · (4.13) = 20.03. Thus, using TS_SS measure, $\boldsymbol{P}$ is more similar to $\boldsymbol{M}$ than $\boldsymbol{Q}$. In Fig. 4.3 (c), $TS\_SS(\boldsymbol{M},\boldsymbol{P})$ = (1) · (5.71) and $TS\_SS(\boldsymbol{M},\boldsymbol{Q})$ = (1) · (2.29) which indicate that $\boldsymbol{Q}$ is more similar to $\boldsymbol{M}$ than $\boldsymbol{P}$. Finaly, for Fig. 4.3 (d), $TS\_SS(\boldsymbol{M},\boldsymbol{P})$ = 1.16 < $TS\_SS(\boldsymbol{M},\boldsymbol{Q})$ = 3.26. These results can consider TS_SS as a more accurate representation of the true similarity between vectors. In order to prevent numerical instability when dealing with very small angles that are extremely close to 0° or 180°, an epsilon (ε) adjustment is typically added to the angle value in both TS and SS leading to $\theta' = \theta + \varepsilon$. In this case, TS_SS will designate a similarity of 0 only when both ED and MD equal 0, indicating identical vectors in both direction and magnitude, signifying the maximum possible similarity between two vectors.

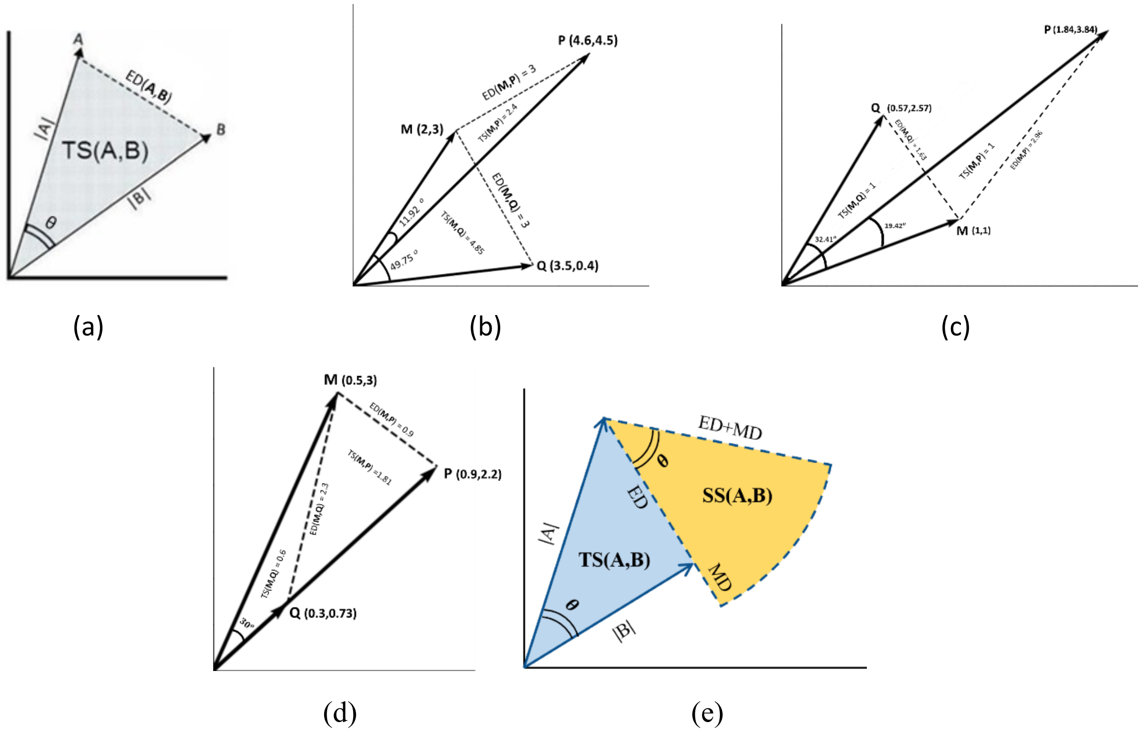


Figure 4.3: Similarity Algorithms. (a) Euclidean Distance and Triangle Similarity. (b) Example of ED drawback. (c) Example of Triangle Similarity drawback. (d) Example of two vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ having different ED and TS to vector $\boldsymbol{M}$. (e) Triangle Similarity-Sector Similarity (TS_SS).

In view of the above discussions and observations, we apply the TS_SS metric as a measure of similarity for matching a query pore with a template pore. Using this measure, the similarity between the feature representation of a pore in $\boldsymbol{PF_Q}$ corresponding to a query image with each of the feature representations of all the pores $\boldsymbol{PF_T}$ corresponding to a template image is computed by following the steps given below:

1. The similarity between the feature representation $\boldsymbol{PF_{Q_i}}$ of the i$^{\text{th}}$ pore in the query image and that of $\boldsymbol{PF_{T_j}}$ of the j$^{\text{th}}$ pore in the template image is computed, using the TS_SS similarity measure given by (4.5) as

$$S_{ij} = TS\_SS\left(\boldsymbol{PF_{Q_i}}, \boldsymbol{PF_{T_j}}\right), \mathrm{i} = 1,2,\dots,\boldsymbol{N};\ \mathrm{j} = 1,2,\dots,\boldsymbol{M} \tag{4.6}$$

2. For the pore features representation $\boldsymbol{PF_{Q_i}}$, there is a set $\{S_{i1}, S_{i2}, \dots, S_{iM}\}$ of M values, corresponding to the M pores in the template image, for the similarity measures. Assuming that S_i^m is the smallest value in this set, we can form a set of the similarity measures of all the pores in the query image as

$$S_m = \begin{bmatrix} S_1^m \\ \vdots \\ S_i^m \\ \vdots \\ S_N^m \end{bmatrix} \tag{4.7}$$

3. The above vector of the minimum similarity values corresponding to N pores in the query image are then averaged to obtain a threshold value δ as

$$\delta = \frac{\sum_{i=1}^{N} S_i^m}{N} \tag{4.8}$$

4. The ith pore in the query image is considered to be matched with the jth pore in the template image if $S_{ij} < \quad \delta$ then the coordinates of the pores pair (i, j) is included in the pore correspondence set C. At the end of the matching process, if K pairs of pores are found to be matched, then C will have K pairs or coordinates, $c_1,\ c_2, \dots, c_K$ **.** The pore correspondence set C is the output of the pore matching segment of Module 2 and is used as the input of Segment 3 of the module for pore matching refinement in order to obtain refined correspondence set $\boldsymbol{RC}$.

4.3 Experimental Results of the Proposed Module 2

In this section, the proposed scheme for pore feature extraction and matching scheme is experimentally validated. First, the details of the dataset used, the procedure used for the training of the proposed network and the figures of merit employed for the performance evaluation of the proposed and the other schemes used for comparison are described. Then, the results of the various experiments performed on the proposed pore-based fingerprint recognition scheme are presented and analyzed, and its performance is compared with that of the existing state-of-the-art schemes.

4.3.1 Training Details and Evaluation Procedure for the Proposed Module 2

In our experiments, images from the PolyU High-Resolution-Fingerprint (HRF) database are used for the training and testing of the proposed network and for the performance evaluation of the proposed fingerprint recognition scheme. The database has two sets of high-resolution (1200 dpi) fingerprint images referred to as DBI and DBII. DBI consists of 210 training images (DBI-training) from 35 fingers each with 6 different impressions, and 1,480 testing images (DBI-test) from 148 fingers each with 10 different impressions. The size of the fingerprint images in the DBI dataset is 320×240 pixels. DBII dataset contains 1,480 images from 148 fingers each with 10 different impressions. The size of the fingerprint images in this dataset is 640×480 pixels and all the images in this dataset are used for testing. The images of the same finger in both sets were collected in two sessions (separated by about two weeks), each capturing 50% of the impressions of the same fingers.

The fingerprint images in the DBI-training dataset are further partitioned randomly into 80% training (query) and 20% validation (template) images. In the proposed network, each pore patch of a finger is classified into one of the n different classes, where n is the maximum number of pores that have been captured in at least one of the impressions of the finger. For a training or test image, a patch of size $p \times p$ ($p = 31$), centered at the centroid of the pore, is formed for each of the pores detected in Module 1. The total number of patches for all the training images combined together in the DBI-training is 85,410 patches. In the testing phase, the network outputs a pore feature representation with a 256-dimensional vector corresponding to a single pore from the query fingerprint image. Once the feature representations of all the pores of the query fingerprint image and a template fingerprint image have been obtained, these representations are used to determine their similarities using pore matching and refinement schemes of Module 2.

For the training of the proposed network, deep supervision [124] is employed in a multi-task learning framework by making the network's parameters learned using a cost function which is the total of the weighted losses making use of the output maps $\boldsymbol{Y}_{FC1}, \boldsymbol{Y}_{FC2}, \boldsymbol{Y}_{FC3}, \boldsymbol{Y}_{FC4}, \boldsymbol{Y}_{FC5}$ at the intermediate layers, as well as the output map $\boldsymbol{Y}_o$ at the final layer,

$$Loss_{total} = \sum_{k=1}^{6} \alpha_k Loss_{id_k} + \beta_k Loss_{vr_k} \tag{4.9}$$

where $Loss_{id_k}$ and $Loss_{vr_k}$ are the identification and verification loss functions at the k^{th} stage, and α_k (β_k) are the weights used for the loss function $Loss_{id_k}$ ($Loss_{vr_k}$) that allow to have a fine-grained control over the relative importance of the identification (verification) loss between the various stages. These weights are set with $\sum_{k=1}^{6} \alpha_k = 1$ ($\sum_{k=1}^{6} \beta_k = 1$) to give the best performance for Module 2.

For the construction of the identification loss function at the kth stage, the map at the kth stage ($\boldsymbol{Y}_{FC1}, \boldsymbol{Y}_{FC2}, \boldsymbol{Y}_{FC3}, \boldsymbol{Y}_{FC4}, \boldsymbol{Y}_{FC5}$ or $\boldsymbol{Y}_o$) is flattened and fed to a fully connected dense layer followed by an n-way Softmax layer [127], which outputs the predicted probability distribution $\widehat{pb}_i$ (i = 1,…,n)over the n classes. The identification loss function is computed using cross-entropy as

$$Loss_{id} = -\sum_{i=1}^{n} pb_i \log \widehat{pb}_i \tag{4.10}$$

where pb_i is the target probability distribution. pb_i = 0 for all i except for i=t, the target class, when pb_t = 1.

The purpose of the verification loss function is to minimize the difference between feature representations of the same pore class and maximize the difference between feature representations of different pores classes. We adopt the loss function proposed by Hadsell *et al.* [128], which uses the L2 norm, and is given by

$$Loss_{vr}\left(\boldsymbol{PF}_{\boldsymbol{Q}_i}, \boldsymbol{PF}_{\boldsymbol{T}_j}, y_{ij}\right) = \begin{cases} \left\|\boldsymbol{PF}_{\boldsymbol{Q}_i} - \boldsymbol{PF}_{\boldsymbol{T}_j}\right\|_2 & if\ y_{ij} = 1 \\ \max\left\{0, \left(m - \left\|\boldsymbol{PF}_{\boldsymbol{Q}_i} - \boldsymbol{PF}_{\boldsymbol{T}_j}\right\|_2\right)\right\} & if\ y_{ij} = -1 \end{cases} \tag{4.11}$$

where y_{ij} is a variable that has values of 1 or -1 depending on wither $\boldsymbol{PF}_{\boldsymbol{Q}_i}$ and $\boldsymbol{PF}_{\boldsymbol{T}_j}$ belong to the same or two different classes, and m is a positive number insuring that the L2 norm to be larger than m when the two pores do not belong to the same class. For our problem, a value of $m = 1$ has been found to be a suitable value for the margin between $\boldsymbol{PF}_{\boldsymbol{Q}_i}$ and $\boldsymbol{PF}_{\boldsymbol{T}_j}$ when the two pores do not belong to the same class.

We employ the stochastic gradient descent (SGD) algorithm with the Nesterov acceleration module [118] to update the parameters and minimize the loss. Data augmentations [116] of the patches through their rotations by 90° and 180°, flipping and Gaussian blurring are utilized to enrich the training dataset and enhance the generalization capability of the network. In the training phase, the initial learning rate and the momentum parameter are set as 0.9 and 0.1, respectively. The learning rate is decreased by 0.01 after every 10 epochs until the 100 epochs after which it is fixed to a value of 0.01. The network is trained with a batch size of 256 and a maximum number of 500 epochs. The experiments are performed in a Python environment on a supercomputer as illustrated in chapter 3 (section 3.4.1).

The figures of merit that are used for the performance evaluation are as follows:

4. Equal error rate (EER) is used to predetermine the threshold values for its false matching rate (FMR) and its false-negative matching rate (FNMR). When the rates are equal, the common value is referred to as the EER. The value indicates that the proportion of false matching is equal to the proportion of false-negative matching. The lower the equal error rate value, the higher the accuracy of the fingerprint recognition system.
5. False matching rate 1000 (FMR1000) indicates the lowest FNMR when FMR ≤ 0.1. The lower the FMR1000 value, the higher the accuracy of the scheme.
6. The quantity of parameters employed within the network and the number of FLOPs denote the computational expense of the CNN network.

We apply the same procedure to evaluate and compare with the proposed scheme as used by other works for fingerprint recognition schemes in the literature. For DBI-test and DBII datasets, each fingerprint image of the same finger in the first session (a total of five images for each finger) is matched against all the five fingerprint images of the same finger in the second session to compute the FNMR with a total of 3700 genuine matches. On the other hand, the first fingerprint image of each finger in the second session is matched against the first fingerprint images of the other fingers from the first session to compute the FMR with a total of 21,756 imposter matches.

4.3.2 Performance Evaluation of the Proposed Scheme

The objective of this subsection is to study the performance of the proposed pore-based fingerprint recognition scheme and compare it with that of other schemes in the literature. We conduct this study in three parts. In the first part, we study the impact of the pore patch size from the fingerprint images on the performance of the proposed module. In the second part, we study the impact of replacing either the proposed network or the proposed pore matching method of the proposed Module 2 with the one used in other neural network-based modules. The computational times of the network Segment (Segment 1) and the pore matching method segment (Segment 2) of the proposed module along with its network complexity, in terms of the number of parameters and the number of FLOPs, are also compared with those of the other network-based modules. In the third part, we compare both the quantitative and qualitative performance of the proposed scheme with those of the other schemes that are neural network-based or otherwise.

4.1 Experimental Results – Part 1

In this subsection of our experiments, we examine the impact of the patch size used by the proposed CNN on the overall performance of our fingerprint recognition scheme. We perform the experiments on the proposed module by using different values of the patch size p, $p = 19, 25, 31\ and\ 41$, corresponding to each of the pores detected from a fingerprint image by Module 1. Table 4.3 lists the EER values of the fingerprint recognition result as well as the number of FLOPs required by the network using the images of DBI-test dataset for the different patch sizes. Fig. 4.4 shows the ROC (FMR vs FNMR) curves for the five different patch sizes. It is seen that the networks with input pore patch sizes of 31×31 and 41×41 pixels provide very close results in terms of ROC performance. It is seen from the results of Table 4.3 and Fig. 4.4 that the increase in the patch size beyond 31×31 results in a modest improvement in the performance of the network with a high computational cost in terms of the number of FLOPs. Therefore, 31×31 is the optimum patch size for the best performance of the proposed module.

Table 4.3: Performance of the proposed module with different pore patch sizes evaluated on the DBI-test dataset.

	Patch size			
	19× 19	25 × 25	31 × 31	41 × 41
EER	4.33	2.23	**1.02**	1.00
Number of FLOPs	2.58M	5.05M	**6.18M**	11.9M

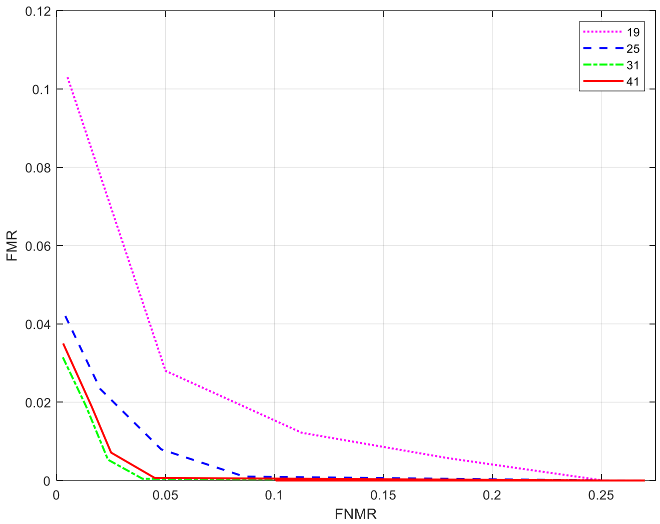


Figure 4.4: ROC curves of the proposed module for different input pore patch sizes.

4.2 Experimental Results – Part 2

In this subsection, we study the impact of using different schemes for Module 1 (Chapter 3) as well as using different schemes for Segments 1 and 2 of Module 2 on the overall performance of a fingerprint recognition system. Table 4.4 provides the results of this study in terms of the performance metric EER. The results in this table are divided into four blocks. In the first block, we present the EER results of our proposed fingerprint recognition scheme (that uses the scheme of Chapter 3 for Module 1 and Segments 1 and 2 of the proposed Module 2) as well as that of the CNN-based fingerprint recognition schemes of [63] and [64]. It is seen from the results of this block of the table that the proposed scheme significantly outperforms the CNN-based schemes of [63] and [64] for both the DBI-test set and DBII test dataset. The second block of the table provides the EER performance in which Module 1 of [63] and [64] has been replaced by the module proposed in Chapter 3 while Module 2 is kept the same as those proposed in [63] and [64]. It is seen from this block of the table that both [63] and [64] can improve their performances by replacing their Module 1 with the proposed pore detection module (Chapter 3) for both the datasets. However, the performance of both these schemes still remains inferior to that of our proposed scheme. The third and fourth blocks of the table present the EER performance, where only Segment 1 or only Segment 2 of Module 2 in [63] and [64], has been replaced by the corresponding segment of our proposed Module 2, while keeping the other segment of Module 2 as well as Module 1 to be the same as those proposed in [63] and [64]. It is seen from the results of these two blocks of Table 4.4 that the schemes of [63] and [64] can again benefit significantly if only one of the two segments, Segment 1 or Segment 2, is replaced by the corresponding segment of our proposed scheme. However, it is noted that even with just modifications in the schemes of [63] and [64], their performance still remains significantly lower than that of our proposed scheme for both the datasets.

From the results of Table 4.4, it can be concluded that the proposed fingerprint recognition system, that consists of the proposed Modules 1 and 2, significantly outperforms the other two CNN-based fingerprint recognition systems, [63] and [64], reported in the literature, and that these two schemes can improve their performance substantially by replacing their segments of Module 2, or Module 1 or 2 totally by the ones used in the proposed scheme for fingerprint recognition system.

Table 4.4: Performance comparison of the proposed scheme and the CNN-based schemes of [63] and [64] on DBI-test set and DBII test dataset without using pore matching refinement.

Module			EER	
Module 1 Fingerprint Pore Detection	Module 2 Pore Feature Extraction and Matching		DBI-test	DBII
	Segment 1 Pore Feature Extraction and Representation Network	Segment 2 Pore Matching		
Proposed Module 1	Proposed	Proposed	**1.02**	**0.094**
[63]	[63]	[63]	2.91	0.57
[64]	[64]	[64]	1.82	1.21
Proposed Module 1	[63]	[63]	2.59	0.53
Proposed Module 1	[64]	[64]	1.69	1.15
[63]	Proposed	[63]	1.98	0.5
[64]	Proposed	[64]	1.48	0.87
[63]	[63]	Proposed	2.51	0.51
[64]	[64]	Proposed	1.62	1.01

Table 4.5 provides the average run times per fingerprint image match of the proposed scheme and that of the schemes of [63] and [64], taken by the respective neural networks (Segment 1 of Module 2) and by the pore matching methods (Segment 2 of Module 2) individually and together, as well as the complexity of the networks in terms of the number of parameters and the number of FLOPs. All the neural network-based modules have been implemented in the same software and hardware environments as stated in Section 3.4.1. It is seen from this table that for both the datasets, the processing time of the neural network of [63] is lower than that of [64] in view of the simplicity of the latter network. However, the network of the proposed module takes significantly less time in comparison to the times taken by either of the other two networks. It is also seen from this table that the proposed pore matching method and the method in [64] take about the same time which is much lower than the time taken by the method in [63]. This is so in view of the fact that the method in [63] requires finding the pores in the template fingerprint image with the best and the second-best match with a pore in the query fingerprint image, while the method in [64] uses a less time-consuming procedure similar to that used by the proposed method. It must also be pointed out that even though the times taken by the proposed pore matching method and pore matching method of [64] are similar, the proposed matching scheme results in an accuracy that is higher than achieved by using the matching scheme of [64]. In order to show this, in our proposed fingerprint recognition system, we replace the pore matching method with the one used in [64] and obtain lower EER values of 1.23 and 0.108 from the resulting system for DBI-test set and DBII test dataset, respectively. Despite the fact that the network used by the proposed scheme has a smaller processing time and provides more accurate performance results, its complexity in terms of the number of parameters utilized and the number of FLOPs consumed by its network is quite impressive. The number of parameters utilized, and the number of FLOPs consumed by the network of the proposed module is only ~33% and ~17%, respectively, of that utilized and consumed by the network of [64], the best performing neural network-based fingerprint recognition system exiting in the literature.

Table 4.5: Comparison of average time per match (in milli-seconds) and network complexity of the proposed scheme and the CNN-based schemes of [62] and [63] on DBI-test set and DBII test dataset.

		[63]	[64]	Proposed module
DBI-test set	Neural Network (Segment 1 of Module 2)	1.95	2.51	**1.38**
	Pore matching method (Segment 2 of Module 2)	2.84	1.08	**1.10**
	Total	4.79	3.59	**2.48**
DBII test datase	Neural Network (Segment 1 of Module 2)	4.61	7.84	**3.99**
	Pore matching method (Segment 2 of Module 2)	5.88	3.49	**3.54**
	Total	10.49	11.33	**7.53**
	Number of Network Parameters	2.99M	875.68k	**291,018**
	Number of FLOPs used by Network	134.62M	35.8M	**6.18M**

4.3 Experimental Results – Part 3

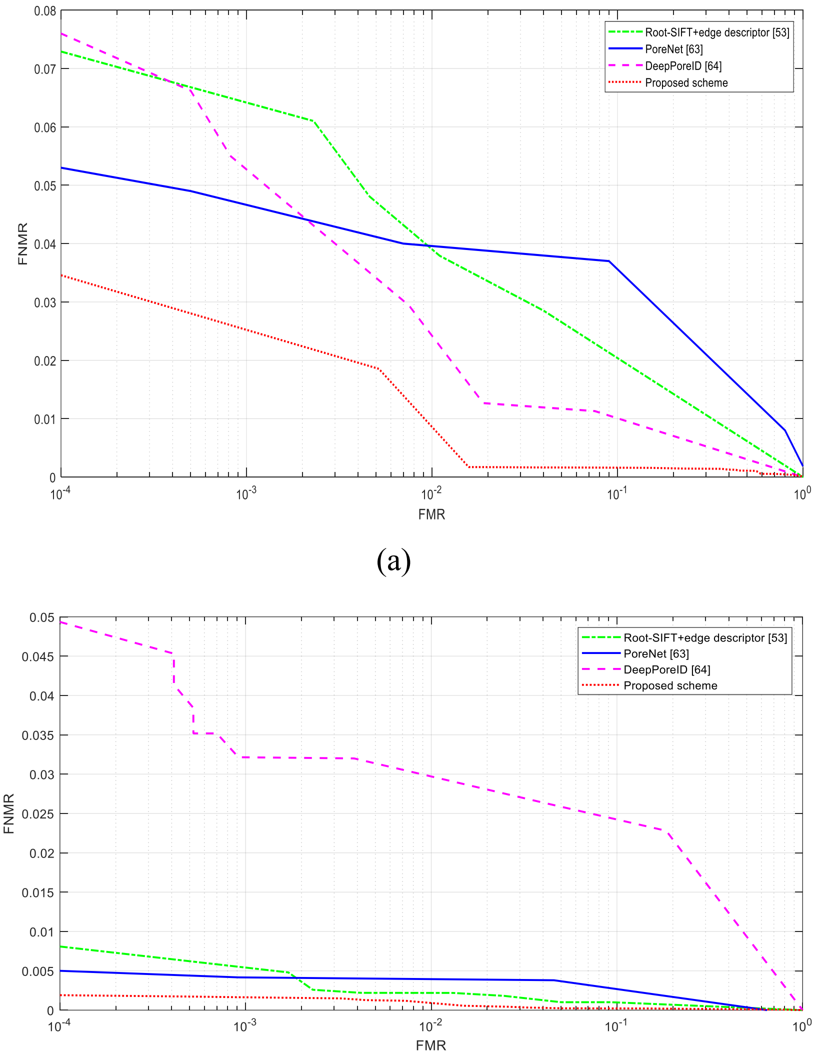
In this subsection of our experimental study, we compare the quantitative and qualitative performance of the proposed pore-based fingerprint recognition system with that of both the traditional and neural network-based schemes. The traditional schemes that are used for comparison are those reported in [38], [50], [51], [52], [53], [71], [73], [129], [130], whereas the neural network-based schemes are the same as those used in Table 4.6, i.e., the schemes of [63] and [64]. Table 4.6 gives the pore-based fingerprint recognition accuracy in terms of EER and the FMR1000 provided by the various schemes for DBI-test set and DBII test dataset. Note that some of the traditional schemes have provided their performance results without pore matching refinement and these schemes are so indicated by marking them with an asterisk (*). It should also be noted that the performance results of our proposed scheme in this table have been provided both without using a pore matching refinement scheme and using three different kinds of pore matching refinement schemes, namely, RANSAC [131], WRANSAC [106] and graph refinement [54]. It is seen from this table that the performance results provided by the proposed scheme, even without using a pore matching refinement scheme significantly outperform all the other existing fingerprint recognition schemes used for comparison in this table.

The performance results of our proposed fingerprint recognition scheme are further improved by using one or the other pore match refinement scheme, with the highest performance results been achieved using the graph refinement scheme. Fig. 4.5 shows the plots of the FNMR vs FMR ROC curves for the proposed scheme and the other two neural network-based schemes of [63] and [64], as well as those for the best performing traditional scheme of [53]. Note that for DBI-test dataset, the ROC curve resulting from the scheme of [63] shows lower FNMR values for the FMR values greater than 0.0021 compared to that for DBII test dataset for which the scheme of [63] provides lower values of FNMR for the entire range of FMR values. However, it is seen from this figure that the proposed scheme provides a lower value for FNMR for any value of FMR compared to that provided by the two CNN-based schemes of [63] and [64] as well as that provided by the traditional scheme of [53] for both the test sets. Therefore, in terms of the ROC curves as well, the proposed scheme can be regarded as having the best performance.

Table 4.6: EER and FMR 1000 performance comparison on the test datasets against state-of-the-art methods.

Module	DBI-test		DBII	
	EER	FMR 1000	EER	FMR 1000
MICP [38]*	30.45	N/A	7.83	N/A
Greedy Matching [129]*	17.67	N/A	0.72	N/A
DP [50]	15.42	N/A	7.05	N/A
Spare Representation +DP [71]	6.59	N/A	0.97	N/A
TDSWR [51]	3.25	11.07	0.53	4.07
SIFT [73]*	3.74	N/A	0.76	N/A
Multi-features + TDSWR [130]	2.17	N/A	0.17	N/A
Data-driven descending algorithm [52]*	1.73	6.93	0.51	2.01
Root-SIFT feature and edge descriptor [53]	1.86	3.92	0.12	0.21
PoreNet [63] without refinement	2.91	5.42	0.57	0.96
DeepPoreID [64] without refinement	1.82	8.83	1.21	9.72
PoreNet [63] + RANSAC [131]	2.27	4.72	0.24	0.4
DeepPoreID [64] + WRANSAC [106]	1.42	7.69	0.51	4.05
Proposed module with no refinement	**1.02**	**1.63**	**0.094**	**0.11**
Proposed module + RANSAC [131]	**0.99**	**1.58**	**0.092**	**0.1**
Proposed module + WRANSAC [106]	**0.97**	**1.44**	**0.09**	**0.095**
Proposed module + Graph refinement [54]	**0.85**	**1.00**	**0.061**	**0.059**

* Traditional schemes without using pore matching refinement.

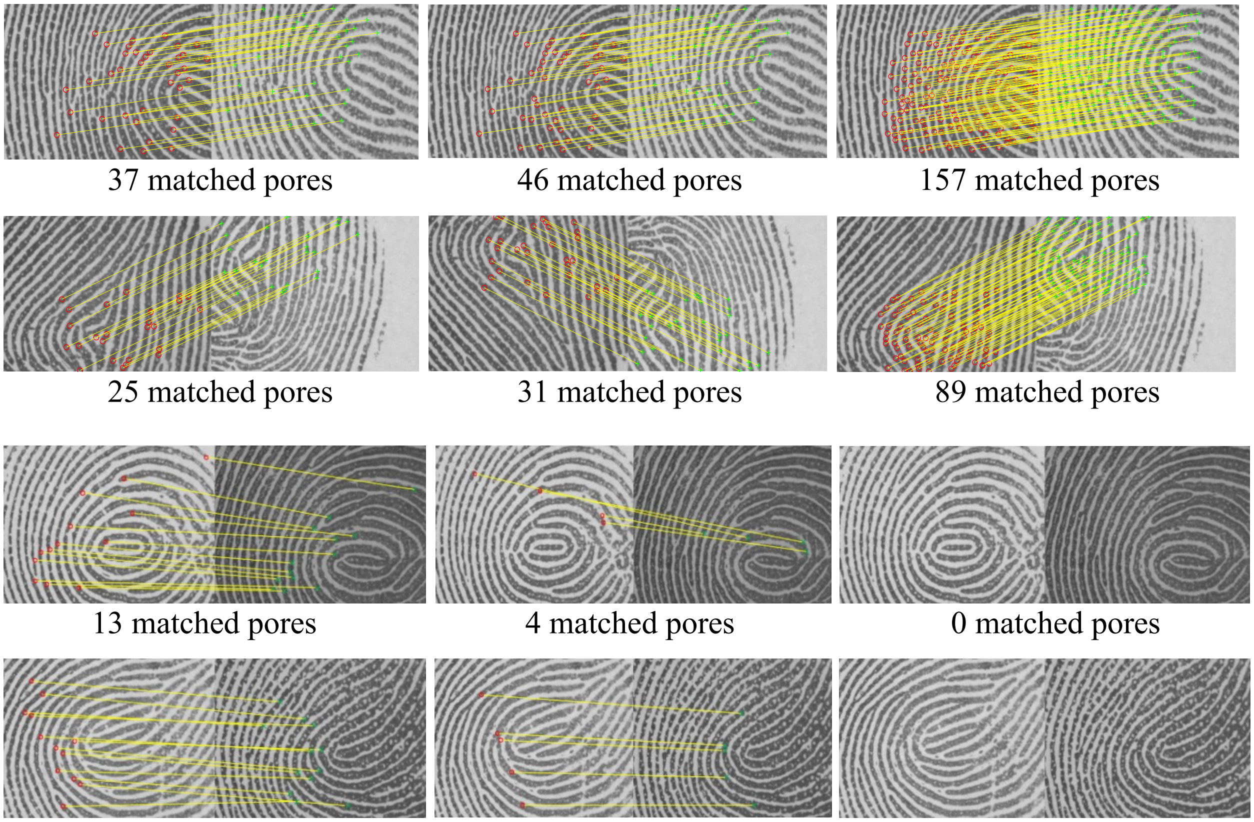


(a)

(b)

Figure 4.5: ROC curves of the proposed scheme and the schemes of *[53]*, *[63]* and *[64]* for (a) DBI_test set, (b) DBII test dataset.

In Fig. 4.6, we provide a visual illustration of pore matching between a pair of genuine fingerprint images and also between a pair of imposter fingerprint images. For the purpose of this illustration, we have selected two pairs of genuine fingerprints (rows 1 and 2 of Fig. 4.6) and two pairs of imposter fingerprints (rows 3 and 4 of Fig. 4.6). Columns 1, 2 and 3 are the results of matching the pores between the pairs of images provided by the schemes of [63] and [64], and the proposed scheme, respectively. It is seen from this figure that in the cases of genuine fingerprints, the proposed scheme is able to match a much larger number of true matching compared to that provided by the other two schemes, where as in the cases of imposter fingerprints, the proposed scheme does not match any of the pores between the pairs of the images and the schemes of [63] and [64] falsely match 11 and 4 pairs of pores, respectively, in one pair, and 13 and 5 pairs of pores, respectively, in the other pair.



37 matched pores

46 matched pores

157 matched pores

25 matched pores

31 matched pores

89 matched pores

13 matched pores

4 matched pores

0 matched pores

12 matched pores

5 matched pores

0 matched pores

Figure 4.6: Matching results of two genuine fingerprint pairs and two imposter fingerprint pairs using PoreNet [62] (left column), DeepPoreID [63] (middle column) and the proposed scheme (right column).

Table 4.7 presents the average numbers of true and false pore matches per pair of genuine and imposter fingerprints, respectively, as obtained by using the proposed scheme and that of using the schemes of [63] and [64]. For genuine pairs of fingerprints, the proposed scheme has a significantly higher number of true pore matches (138 for DBI-test set and 951 for DBII test dataset) compared to that obtained by using the two schemes. At the same time, for imposter pairs of fingerprints, the proposed scheme has a much lower number of false pore matches (4 for DBI-test set and 6 for DBII test dataset) compared to that obtained using the schemes of [63] and [64]. These results again support the superior capability of the proposed scheme in fingerprint recognition.

Table 4.7: Average number of true and false pore matches per pair of genuine and imposter fingerprints.

The average number per pair	Test Dataset	PoreNet [63]	DeepPoreID [64]	Proposed scheme
True matches in genuine fingerprints	DBI-test	57	82	**138**
	DBII	692	349	**951**
False matches in imposter fingerprints	DBI-test	33	25	**6**
	DBII	12	21	**4**

4.4 Summary

In this chapter, a low complexity, high performance pore feature extraction and matching module has been proposed. The module consists of three segments, a pore feature and representation segment, a pore matching segment, and a pore matching refinement segment. In the first segment, a neural network that generates a hierarchical features by employing depthwise and depthwise separable convolutional layers in a residual framework to produce a highly representable pore features has been designed. In the second segment, a new metric for measuring the similarity between the pore representations in the query and template fingerprint images has been developed. This metric is capable of measuring the differences between the two pores more accurately by incorporating the Euclidean distance, angle, and magnitude differences metrices in a composite metric. In the third segment, a graph-based refinement method, LTPM, has been deployed by removing falsely matched pair of pores from the pore correspondence list. Empirical examination of the proposed module has been conducted using PolyU High-Resolution-Fingerprint database. The experimental results have demonstrated the superiority of the proposed module in terms of the recognition accuracy and network complexity over the traditional and to other CNN-based schemes.

Chapter 5

Conclusion

5.1 Concluding Remarks

Fingerprints are one of the most widely used biometric traits in view of it significant advantages such as high security, affordability, and non-invasiveness. The use of fingerprints as identity evidence can be dated back to hundreds of years. Automated fingerprint recognition systems (AFRSs) have rapidly developed due to the advancement of computing power, fingerprint acquisition technology, and fingerprint processing algorithms. AFRSs are now deployed in both forensics and non-forensic applications. The capacity of fingerprints to distinguish different persons depends highly on the discriminative features that can be extracted from the captured fingerprint images.

Fingerprint features are generally divided into three levels, known as level-1, level-2 and level-3 features, according to their scales. level-1 features refer to the ridge patterns. Level-2 features refer to the minutiae points, such as bifurcations and ending of the ridges. Level-3 features refer to the fine details associated with the ridges, such as ridge contour, pore, dots and incipient ridges. Although level-1 and level-2 features have been commonly used for fingerprint recognition, the distinctiveness of level-1 features is not sufficient for the recognition of the fingerprint and the accuracy of the level-2 features for fingerprint recognition systems are compromised in view of the presence of only small number of minutiae points present in a fingerprint, especially in a partial fingerprint.

In recent years, attention has been paid to the use of pore features (leve-3 features) for fingerprint recognition. This is in view of the fact that in the fingerprint, pore exist in large quantities and the advancement in the high-resolution imaging technology has made capturing of these pores feasible in fingerprints. Recent research efforts in pore-based fingerprint recognition systems have shifted towards utilizing neural networks, particularly CNNs, in view of their capability of automatic generation and extraction of pore features. The integration of neural networks into pore-based fingerprint recognition systems holds promise for diverse real-world applications. However, the challenge lies in balancing accuracy with complexity in designing the network architectures for the purpose of fingerprint recognition. In addition, the available knowledge on the fingerprint pores such as variations in the statistics of the pore intensities from region to region, have not been taken into consideration and even the knowledge that is taken into consideration have not been used optimally. To address these challenges, in this thesis, a CNN-based pore fingerprint recognition system, consisting of a pore detection module and a pore feature extraction and matching module, has been developed. These modules leverage the design of a new neural network architectures, incorporate pore-specific knowledge to detect the pore centroids and develop and utilize a new metric to measure the similarity of the pores accurately.

In the pore detection module of the proposed CNN-based pore fingerprint recognition system, the process of the proposed pore detection has been divided into two parts. In the first part, a neural network architecture has been developed to generate and extract low, middle and high level hierarchical features in a residual framework and then to fuse them to obtain a very rich set of pore features that lead to a highly representational pore intensity map. In this part, the goals of high-performance and low complexity have been achieved by designing the network based on depthwise convolutional layers. The focus in the second part of the pore detection scheme has been on the efficient use of the existing knowledge on fingerprint pores, such as the pore intensity, their variations from one region to another region of the fingerprint image and the minimum distance between two neighboring pores, in order to accurately determine the pore centroids from the pore intensity map.

In the pore feature extraction and matching module of the proposed CNN-based pore fingerprint recognition system, a neural network architecture has been designed to generate hierarchical features efficiently, achieving a balance between performance and complexity using depthwise and depthwise separable convolutional layers. Then, a more accurate composite metric, encompassing the Euclidean distance, angle, and magnitudes difference between the two vectors representing the pore representations, has been introduced to measure the similarity between the pores of the query and template fingerprint images.

The proposed CNN-based pore fingerprint recognition scheme has been extensively experimented on the fingerprint images of the PolyU High-Resolution-Fingerprint database. The performance of the proposed scheme has been evaluated in terms of true and false pore detection rates, EER, FMR1000, the subjective quality of the detected pores as well as of the matched pores between the query and template fingerprints. The experimental results have demonstrated that each part of the first module and each segment of the second module have a significant impact on the performance of the proposed fingerprint recognition system. It has been seen that the performance of the proposed scheme is significantly superior to that of both the traditional and the state-of-the-art CNN-based schemes. Finally, it is worth mentioning that the highly superior performance of the proposed scheme has been achieved with the number of the parameters and the number of arithmetic operations employed by both the pore detection network and the pore feature extraction and representation network are only small fractions of those employed by the CNN networks used in other state-of-the-art CNN-based fingerprint recognition systems.

5.2 Scope for Further Investigation

The primary objective in designing the CNNs in this thesis is to learn hierarchical pore features, enabling the network to extract complex patterns and structures of the pores in the fingerprint images. Moreover, the use of the depthwise and depthwise separable convolution operations in designing the residual blocks of the proposed networks have significantly contributed to the complexity of the proposed networks and the overall accuracy of the proposed fingerprint recognition system. In this context, study can be undertaken to explore other architectures, such as graph neural networks [132] (GNNs), in the context of fingerprint recognition. GNNs are a class of neural networks that operate on graph-structured data, making them well-suited for tasks involving relational data. Since the data used for fingerprint recognition involve relational data, it would be worth investigating GNNs for the extraction and matching of pore features in fingerprints.

The concept of triangle and sector similarity for pore matching in fingerprint recognition systems has been used in this thesis to measure the similarity of the pore representations between the query and template fingerprint images. This metric merges the Euclidean distance, angle and magnitudes difference of vectors representing the pore features. In our proposed system of fingerprint recognition, matching of the pores has been carried out outside the network, which provides pore feature representation. It would be worth investigating a modification of the architecture of the proposed network that carries out the task of pore feature extraction and representation, so that the pore matching is done by the network itself. In this case, one can design a cost function that minimizes the distance between the genuine pores and maximizes the distance between the imposter pores using the proposed composite metric and employ such a cost function for the training for the network.

While the use of graph-based refinement techniques for pore matching refinement has provided enhanced accuracy for the fingerprint recognition system, there remains scope for exploring alternative refinement strategies that could possibly result in further improving the recognition accuracy. One avenue would be to investigate the integration of a machine learning algorithm, such as reinforcement learning [133], into the pore matching refinement process. These algorithms can be trained to learn discriminative patterns and decision rules from large-scale fingerprint datasets, enabling a more accurate identification and filtering of falsely matched pore pairs.

References

[1] J. Anil, L. Hong, and P. Sharath, “Biometric identification” *Commun. of the ACM*, vol. 43, no. 2, pp. 90–98, Feb. 2000.

[2] N. K. Ratha, J. H. Connell, and R. M. Bolle, “Enhancing security and privacy in biometrics-based authentication systems,” *IBM Syst. J.*, vol. 40, no. 3, pp. 614–634, 2001.

[3] J. L. Wayman, “Fundamentals of biometric authentication technologies,” *Int. J. Image Graph.*, vol. 1, no. 01, pp. 93–113, 2001.

[4] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, “Introduction,” in *Handbook of fingerprint recognition*, 3rd ed. London, UK, Springer Verlag, 2009, ch. 1, sec. 1-15, pp. 1-47.

[5] A. C. Weaver, “Biometric authentication,” *Computer,* vol. 39, no. 2, pp. 96–97, Feb. 2006.

[6] S. K. Sahoo, T. Choubisa, and S. M. Prasanna, “Multimodal biometric person authentication: A review,” *IETE Tech. Rev.*, vol. 29, no. 1, pp. 54–75, 2012.

[7] A. Kale *et al.*, “Identification of Humans Using Gait,” *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1163–1173, Sep. 2004.

[8] N. Cuntoor, A. Kale, and R. Chellappa, “Combining multiple evidences for gait recognition,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing,* Hong Kong, China, May 6-10, 2003, pp. 113-116.

[9] D. T. Toledano, R. Fernández Pozo, Á. Hernández Trapote, and L. Hernández Gómez, “Usability evaluation of multi-modal biometric verification systems,” *Interact. Comput.*, vol. 18, no. 5, pp. 1101–1122, Sep. 2006.

[10] J. M. H. Ali and A. E. Hassanien, “An Iris Recognition System to Enhance E-security Environment Based on Wavelet Theory,” *Adv. Model. and Optim.*, vol. 5, no. 2, 2003.

[11] Srinivasa R., “Global Biometrics Market Extended by Affective Computing,” www.bccresearch.com. BCC Research. https://www.bccresearch.com/market-research/information-technology/biometrics-technologies-markets-report.html (Jan. 02, 2023).

[12] R. Mahajan, T. Gupta, S. Mahajan, and N. Bawa, “Retina as Authentication Tool for Covert Channel Problem,” *World Acad. of Sci. Eng. and Technol.*, vol. 3, no. 8, pp. 1517–1522, 2009.

[13] Klaus Schmeh, "Authentication," in *Cryptography and public key infrastructure on the Internet*. England, UK, Wiley, 2003, ch. 13, sec. 1-3, pp. 179-191.

[14] J. R. Vacca, "Biometric technologies for personal identification," in *Biometric technologies and verification systems*. Oxford, UK, Elsevier Science and Technology, 2007, ch. 2, pp. 24.

[15] W. A. W. Adnan, L. T. Siang, and S. Hitam, "Fingerprint recognition in wavelet domain," *Jurnal Teknologi*, 41(D), Universiti Teknologi Malaysia, pp. 25-42, Dis. 2004.

[16] A. K. Jain, "Introduction to Biometrics" *in Handbook of Biometrics*. New York, NY, USA, Springer Science and Business Media, 2008, ch. 1, p. 1-22.

[17] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE Secur. Priv.*, vol. 1, no. 2, pp. 33–42, Apr. 2003.

[18] A. K. Jain and S. Z. Li, "Face Image Modeling and Representation" in *Handbook of face recognition*, London, UK, Springer-Verlag, 2011, ch. 1, pp-19-276.

[19] M. S. Arya, N. Jain, J. Sisodia, and N. Sehgal, "DNA encoding based feature extraction for biometric watermarking," in *Proc. Int. Conf. on Image Inf. Process.*, Shimla, India, Nov. 3-5, 2011, pp. 1–6.

[20] A. K. Jain, A. Ross, and U. Uludag, "Biometric template security: Challenges and solutions," in *Proc. 13th European signal processing conference*, Antalya, Turkey, Sep 4-9, 2005, pp. 1–4.

[21] C. Meena, "Fuzzy Image Processing for Offline Signature Verification and Interpretation," in *Proc. IPCV*, Las Vegas Nevada, USA, July 14-17, 2008, pp. 544–549.

[22] J. Daugman, "Probing the uniqueness and randomness of iris codes: Results from 200 billion iris pair comparisons," in *Proc. IEEE*, vol. 94, no. 11, Nov. 2006, pp. 1927–1935.

[23] M. Rybnik, M. Tabedzki, and K. Saeed, "A keystroke dynamics based system for user identification," in *Proc. 7th computer information systems and industrial management applications*, Ostrava, Czech Republic, June 26-28, 2008, pp. 225–230.

[24] K. Moustakas, D. Tzovaras, and G. Stavropoulos, "Gait recognition using geometric features and soft biometrics," *IEEE Signal Process. Lett.*, vol. 17, no. 4, pp. 367–370, Apr. 2010.

[25] J. D. Woodard Jr, N. M. Orlans, and P. T. Higgins, *Biometrics: Identity Assurance in the Information Age*. New York: McGraw-Hill/Osborne, 2003.

[26] U. Halici, L. C. Jain, and A. Erol, "Introduction to fingerprint recognition," in *Intelligent biometric techniques in fingerprint and face recognition*, Boca Raton, FL, USA, Routledge, 2022, pp. 1–34.

[27] M. Dharchaudhuri, "Indexing of Large Biometric Database," B.S. thesis, Department of Computer Science and Engineering, National Institute of Technology, Rourkela, 2010.

[28] R. Kwok, "Fake finger reveals the secrets of touch," Nature Publishing Group, vol. 29, 2009. Accessed: Sep. 23, 2019. [Online]. Available: http://www.nature.com/news/2009/090129/full/news.2009.68.html,

[29] R. D. Olsen, "The chemical composition of palmar sweat," *Fingerpr. Identif. Mag.*, vol. 53, no. 10, pp. 3–23, 1972.

[30] G. L. Marcialis and F. Roli, "Fusion of LDA and PCA for Face Recognition," Dep. Electr. Electron. Eng. Univ. Cagliari Piazza DíArmi, 2002.

[31] A. M. Padmanaban, *Role of Fingerprint Science in Police and Judiciary*, A.M. Padmanaban, 2008. Accessed: Mar. 12, 2019. [Online]. Available: https://books.google.ca/books?id=7IsxQwAACAAJ

[32] P. L. Johnson, "Life of Latents," *Fingerpr. Identif. Mag.*, vol. 54, no. 3, pp. 11–15, 1972.

[33] A. K. Datta, H. C. Lee, R. Ramotowski, and R. E. Gaensslen, *Advances in fingerprint technology*, 2nd ed., USA, CRC press, 2001.

[34] C. Champod, C. J. Lennard, P. Margot, and M. Stoilovic, *Fingerprints and other ridge skin impressions*. CRC press, 2004, ch. 1-4, pp. 1-176.

[35] T.-Y. Jea, V. Govindaraju, "A Minutiae-based partial fingerprint recognition system," State University of New York at Buffalo, 2005.

[36] P. Pathak, "Image compression algorithms for fingerprint system," *Pattern Recognition*, vol. 38, no. 10, pp. 1672–1684, Oct. 2005.

[37] K. Mieloch, A. Munk, and P. Mihailescu, "Hierarchically linked extended features in fingerprints," in *Proc. Biometrics Symposium*, Tampa, FL, US, Sep. 23-25, 2008, pp. 47–52.

[38] A. K. Jain, Y. Chen, and M. Demirkus, "Pores and ridges: High-resolution fingerprint matching using level 3 features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 15–27, 2006.

[39] D. R. Ashbaugh, *Quantitative-qualitative friction ridge analysis: an introduction to basic and advanced ridgeology*, USA, CRC press, 1999.

[40] D. Zhang, F. Liu, Q. Zhao, G. Lu, and N. Luo, “Selecting a reference high resolution for fingerprint recognition using minutiae and pores,” *IEEE Trans. Instrum. Meas.*, vol. 60, no. 3, pp. 863–871, 2010.

[41] P. Johnson and S. Schuckers, “Fingerprint pore characteristics for liveness detection,” in *Proc. International Conference of the Biometrics Special Interest Group (BIOSIG)*, Darmstadt, Germany, Sep. 10-12, 2014, pp. 1–8.

[42] B. Bindra, O. P. Jasuja, and A. K. Singla, “Poroscopy: A method of personal identification revisited,” *Anil Aggrawals Internet J. Forensic Med. Toxicol.*, vol. 1, no. 1, 2000.

[43] L. Nanni and A. Lumini, “Local binary patterns for a hybrid fingerprint matcher,” *Pattern Recognit.*, vol. 41, no. 11, pp. 3461–3466, 2008.

[44] Q. Zhao, D. Zhang, L. Zhang, and N. Luo, “High resolution partial fingerprint alignment using pore–valley descriptors,” *Pattern Recognit.*, vol. 43, no. 3, pp. 1050–1061, 2010.

[45] H. Choi, K. Choi, and J. Kim, “Fingerprint matching incorporating ridge features with minutiae,” *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 2, pp. 338–345, 2011.

[46] N. Ratha and R. Bolle, *Automatic fingerprint recognition systems*. New York, USA, Springer Science & Business Media, 2003.

[47] J. D. Stosz and L. A. Alyea, “Automated system for fingerprint authentication using pores and ridge structure,” in *Proc. Automatic systems for the identification and inspection of humans*, San Diego, CA, USA, Oct. 1994, pp. 210–223.

[48] K. Kryszczuk, A. Drygajlo, and P. Morier, “Extraction of level 2 and level 3 features for fragmentary fingerprints,” in *Proc. 2nd COST275 Workshop*, 2004, pp. 1-6.

[49] Q. Zhao, D. Zhang, L. Zhang, and N. Luo, “Adaptive fingerprint pore modeling and extraction,” *Pattern Recognit.*, vol. 43, no. 8, pp. 2833–2844, 2010.

[50] Q. Zhao, L. Zhang, D. Zhang, and N. Luo, “Direct pore matching for fingerprint recognition,” in *Proc. International Conference on Biometrics*, 2009, pp. 597–606.

[51] F. Liu, Q. Zhao, and D. Zhang, “A novel hierarchical fingerprint matching approach,” *Pattern Recognit.*, vol. 44, no. 8, pp. 1604–1613, 2011.

[52] Y. Xu, G. Lu, Y. Lu, F. Liu, and D. Zhang, “Fingerprint pore comparison using local features and spatial relations,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 10, pp. 2927–2940, 2018.

[53] Y. Xu, G. Lu, Y. Lu, and D. Zhang, “High resolution fingerprint recognition using pore and edge descriptors,” *Pattern Recognit. Lett.*, vol. 125, pp. 773–779, 2019.

[54] Y. Xu, Y. Lu, G. Lu, J. Li, and D. Zhang, "Fast Pore Comparison for High Resolution Fingerprint Images Based on Multiple Co-Occurrence Descriptors and Local Topology Similarities," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 9, pp. 5721–5731, Dec. 2019.

[55] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[56] B. Li and Y. He, "An improved ResNet based on the adjustable shortcut connections," *IEEE Access*, vol. 6, pp. 18967–18974, 2018.

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

[58] 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.

[59] S.-J. Lee, T. Chen, L. Yu, and C.-H. Lai, "Image classification based on the boost convolutional neural network," *IEEE Access*, vol. 6, pp. 12755–12768, Jan. 2018.

[60] R. D. Labati, A. Genovese, E. Muñoz, V. Piuri, and F. Scotti, "A novel pore extraction method for heterogeneous fingerprint images using convolutional neural networks," *Pattern Recognit. Lett.*, vol. 113, pp. 58–66, 2017.

[61] H.-U. Jang, D. Kim, S.-M. Mun, S. Choi, and H.-K. Lee, "DeepPore: fingerprint pore extraction using deep convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1808–1812, 2017.

[62] V. Anand and V. Kanhangad, "Pore detection in high-resolution fingerprint images using deep residual network," *J. Electron. Imaging*, vol. 28, no. 2, pp. 020502(1-4), 2019.

[63] V. Anand and V. Kanhangad, "PoreNet: CNN-Based Pore Descriptor for High-Resolution Fingerprint Recognition," *IEEE Sens. J.*, vol. 20, no. 16, pp. 9305–9313, Aug. 2020.

[64] F. Liu, Y. Zhao, G. Liu, and L. Shen, "Fingerprint pore matching using deep features," *Pattern Recognit.*, vol. 102, pp. 1-12, Jun. 2020.

[65] Q. Zhao and A. K. Jain, "On the utility of extended fingerprint features: A study on pores," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, Francisco, CA, USA, Jun. 13-18, 2010, pp. 9–16.

[66] Q. Zhao, F. Liu, L. Zhang, and D. Zhang, “Parallel versus hierarchical fusion of extended fingerprint features,” in *Proc. 20th International Conference on Pattern Recognition*, Istanbul, Turkey, Aug. 23-26, 2010, pp. 1132–1135.

[67] H. C. Van Tilborg and S. Jajodia, “Fingerprint,” in *Encyclopedia of cryptography and security*, Springer Science & Business Media, 2014, pp. 460-465.

[68] K. Kryszczuk, A. Drygajlo, and P. Morier, “Extraction of level 2 and level 3 features for fragmentary fingerprint comparison,” *EPFL*, vol. 3, pp. 45–47, 2008.

[69] B. M. Mehtre, “Fingerprint image analysis for automatic identification,” *Mach. Vis. Appl.*, vol. 6, pp. 124–139, 1993.

[70] K. M. Kryszczuk, P. Morier, and A. Drygajlo, “Study of the distinctiveness of level 2 and level 3 features in fragmentary fingerprint comparison,” in *International Workshop on Biometric Authentication*, 2004, pp. 124–133.

[71] F. Liu, Q. Zhao, L. Zhang, and D. Zhang, “Fingerprint pore matching based on sparse representation,” in *Proc. 20th International Conference on Pattern Recognition*, Istanbul, Turkey, Aug. 23-26, 2010, pp. 1630–1633.

[72] R. de Paula Lemes, M. P. Segundo, O. R. Bellon, and L. Silva, “Dynamic pore filtering for keypoint detection applied to newborn authentication,” in *Proc. 22nd International Conference on Pattern Recognition*, Stockholm, Sweden, Aug. 24-28, 2014, pp. 1698–1703.

[73] M. Pamplona Segundo and R. de Paula Lemes, “Pore-based ridge reconstruction for fingerprint recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 128–133.

[74] V. Anand and V. Kanhangad, “Pore based indexing for high-resolution fingerprints,” in *Proc. IEEE international conference on identity, security and behavior analysis (ISBA)*, New Delhi, India, Feb. 22-24, 2017, pp. 1–6.

[75] 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.

[76] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning (PMLR)*, 2013, pp. 1310–1318.

[77] J. Zhang, T. He, S. Sra, and A. Jadbabaie, “Why gradient clipping accelerates training: A theoretical justification for adaptivity,” 2019. *arXiv-1905.11881.*

[78] S. Haykin, *Neural networks and learning machines,* 3rd ed., New Jersey, USA, Pearson Education, 2009.

[79] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Networks: Modern Practices,” in *Deep learning*. London, England, UK, MIT press, 2016, ch. 2, sec. 6-12, pp. 161-474.

[80] R. Donida Labati, A. Genovese, E. Munoz Ballester, V. Piuri, F. Scotti, and G. Sforza, “Computational intelligence for biometric applications: a survey,” *Comput. Int. Sci. J.*, vol. 15, no. 1, pp. 40–49, 2016.

[81] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” 2014, *ArXiv-1404.2188*.

[82] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[83] C. Ding and D. Tao, “Robust face recognition via multimodal deep face representation,” *IEEE Trans. Multimed.*, vol. 17, no. 11, pp. 2049–2058, 2015.

[84] N. Liu, H. Li, M. Zhang, J. Liu, Z. Sun, and T. Tan, “Accurate iris segmentation in non-cooperative environments using fully convolutional networks,” in *Proc. International Conference on Biometrics (ICB)*, Halmstad, Sweden, Jun. 13-16, 2016, pp. 1–8.

[85] N. Liu, M. Zhang, H. Li, Z. Sun, and T. Tan, “DeepIris: Learning pairwise filter bank for heterogeneous iris verification,” *Pattern Recognit. Lett.*, vol. 82, pp. 154–161, 2016.

[86] L. Jiang, T. Zhao, C. Bai, A. Yong, and M. Wu, “A direct fingerprint minutiae extraction approach based on convolutional neural networks,” in *Proc. International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, Jul. 24-29, 2016, pp. 571–578.

[87] D. Menotti *et al.*, “Deep representations for iris, face, and fingerprint spoofing detection,” *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 4, pp. 864–879, 2015.

[88] P. Silva, E. Luz, R. Baeta, H. Pedrini, A. X. Falcao, and D. Menotti, “An approach to iris contact lens detection based on deep image representations,” in *Proc. 28th SIBGRAPI Conference on Graphics, Patterns and Images*, Salvador, Brazil, Aug. 26-29, 2015, pp. 157–164.

[89] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, “Fingerprint liveness detection using convolutional neural networks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 6, pp. 1206–1213, 2016.

[90] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *ArXiv-1409.1556*.

[91] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[92] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[93] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[94] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proc. IEEE conference on computer vision and pattern recognition*, Miami, FL, USA, Jun. 20-25, 2009, pp. 248–255.

[95] T.-Y. Lin *et al.*, “Microsoft coco: Common objects in context,” in Proc. *13th European Conference Computer Vision (ECCV)*, Zurich, Switzerland, Sep. 6-12, 2014, pp. 740–755.

[96] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[97] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, “The importance of skip connections in biomedical image segmentation,” in *Proc. International Workshop on Deep Learning in Medical Image Analysis*, 2016, pp. 179–187.

[98] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[99] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, “Deeply-supervised nets,” *Artificial intelligence and statistics*, vol. 19, pp. 6-9, 2015.

[100] A. A. Abbood, G. Sulong, and S. U. Peters, “A review of fingerprint image pre-processing,” *J. Teknol.*, vol. 69, no. 2, pp. 79–84, 2014.

[101] G. Jr, “AUC-ROC of a random classifier,” Data Science Stack Exchange. https://datascience.stackexchange.com/questions/31872/auc-roc-of-a-random-classifier/31877#31877 (Jan. 0.2 2020)

[102] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, “Euclidean distance matrices: essential theory, algorithms, and applications,” *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 12–30, 2015.

[103] C. Moler and D. Morrison, “Replacing square roots by Pythagorean sums,” *IBM J. Res. Dev.*, vol. 27, no. 6, pp. 577–581, 1983.

[104] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[105] H. Cantzler, “Random sample consensus (RANSAC),” *Inst. Percept. Action Behav. Div. Inform. Univ. Edinb.*, vol. 3, 1981.

[106] D. Zhang, W. Wang, Q. Huang, S. Jiang, and W. Gao, “Matching images more efficiently with local descriptors,” in *Proc. 19th International Conference on Pattern Recognition*, Tampa, FL, USA, Dec. 8-11, 2008, pp. 1–4.

[107] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, “Locality preserving matching,” *Int. J. Comput. Vis.*, vol. 127, pp. 512–531, 2019.

[108] M. Ali, C. Wang, and M. O. Ahmad, “An Efficient Convolutional Neural Network for Fingerprint Pore Detection,” *IEEE Trans. Biom. Behav. Identity Sci.*, vol. 3, no. 3, pp. 332-346, March 2021.

[109] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015, *ArXiv-1502.03167*.

[110] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proc. of ICML*, 2010, pp. 1-8.

[111] W. Niblack, *An introduction to digital image processing*, Englewood Cliffs, NJ, Prentice Hall, 1985.

[112] N. B. Rais, M. S. Hanif, and I. A. Taj, “Adaptive thresholding technique for document image analysis,” in *Proc. 8th Int. Multitopic Conf*, Dec. 24-26, 2004, pp. 61–66.

[113] (2009) PolyU HRF Database, The Hong Kong Polytechnic University. [Online]. Available: http://www4.comp.polyu.edu.hk/~biometrics/HRF/HRF_old.htm

[114] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Nashville, TN: John Wiley & Sons, 2012.

[115] L. Antiga, Retina blood vessel segmentation with a convolution neural network (U-net). Accessed: Sep. 22, 2017. [Online]. Available: https://github.com/orobix/retina-unet

[116] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, “Differential data augmentation techniques for medical imaging classification tasks,” in *Proc. AMIA Annu. Symp.*, vol. 2017, 2017, pp. 979–984.

[117] K. P. Murphy, The Machine Learning: A Probabilistic Perspective. Cambridge, MA, USA: MIT Press, 2014

[118] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *ArXiv-1609.04747*.

[119] H. T. Le *et al.*, "Study of CNN Based Classification for Small Specific Datasets," in *Modern Approaches for Intelligent Information and Database Systems*, (Studies in Computational Intelligence), vol. 769. Cham, Switzerland: Springer, 2018.

[120] Y. Zhu, Q. Ouyang, and Y. Mao, "A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy," *BMC Bioinformatics*, vol. 18, no. 1, pp. 1–10, 2017.

[121] M. Dvornik, "Learning with Limited Annotated Data for Visual Understanding," Ph.D. dissertation, Dept. Mathematics and computer science, Université Grenoble Alpes, France, 2019.

[122] M. Ali, C. Wang, and M. O. Ahmad, "A Deep CNN-Based Feature Extraction and Matching of Pores for Fingerprint Recognition." *IEEE Transaction on Biometrics. Behavior and Identity Science*, 2024 (Submitted)

[123] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," 2014, *ArXiv-1406.4773*.

[124] M. Lin, Q. Chen, and S. Yan, "Network In Network." 2014, *ArXiv-1312.4400*

[125] P. D. Lax, "Linear Mappings," in *Linear algebra and its applications*, 2nd ed., New Jersey, USA, John Wiley & Sons, 2007, ch. 3, pp. 29-30.

[126] A. Heidarian and M. J. Dinneen, "A Hybrid Geometric Approach for Measuring Similarity Level Among Documents and Document Clustering," in *Proc. of IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, Oxford, UK, Mar. 2016, pp. 142–151.

[127] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing: Algorithms, architectures and applications*, Springer, 1990, pp. 227–236.

[128] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA, Jun. 17-22, 2006, pp. 1735–1742.

[129] J. Feng, "Combining minutiae descriptors for fingerprint matching," *Pattern Recognit.*, vol. 41, no. 1, pp. 342–352, 2008.

[130] F. Liu, Y. Zhao, and L. Shen, "Feature guided fingerprint pore matching," in *Proc. of Chinese Conference on Biometric Recognition*, Springer, 2017, pp. 334–343.

[131] L. Qin and W. Gao, "Image matching based on a local invariant descriptor," in *Proc. of IEEE International Conference on Image Processing*, Genova, Italy, Sep. 14, 2005, pp. 1-4.

[132] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[133] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3–4, pp. 219–354, Jun. 2018.