ON THE GENERALIZATION OF FINGERPRINT EMBEDDINGS

By

Steven A. Grosz

A DISSERTATION

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

Computer Science—Doctor of Philosophy

2024

ABSTRACT

Fingerprint recognition is a long-standing and important topic in computer vision and pattern recognition research, supported by its diverse applications in real-world scenarios such as access control, consumer products, law enforcement, forensics, national identity, and border security. Recent advances in deep learning have greatly enhanced fingerprint recognition accuracy and efficiency alongside traditional hand-crafted fingerprint recognition methods, particularly in controlled settings. While state-of-the-art fingerprint recognition methods excel in controlled scenarios, like rolled fingerprint recognition, their performance tends to drop in uncontrolled settings, such as latent and contactless fingerprint recognition. These scenarios are often characterized by extreme degradations and image variations in the captured images. This performance drop is due to the inability of fingerprint embeddings (feature vectors obtained via deep networks) to generalize across variations in the captured fingerprint images between controlled and uncontrolled settings.

The challenges in the generalization of fingerprint embeddings, from controlled to uncontrolled settings, encompass issues such as insufficient labeled data, varying domain characteristics (often referred to as “domain gap"), and the misalignment of fingerprint features due to information loss. This dissertation proposes a series of methods aimed at addressing these challenges in various unconstrained fingerprint recognition scenarios. We begin in chapter 2 with an examination of cross-sensor and cross-material presentation attack detection (PAD), where the sensing mechanism and encountered presentation attack instruments (PA) may be unknown. We present methods to augment the given training data to include a wider diversity of possible domain characteristics, while simultaneously encouraging the learning of domain-invariant representations. Next, we turn our attention in chapter 3 to the challenging scenario of contact to contactless fingerprint matching, where misaligned fingerprint features due to differences in contrast, perspective differences, and non-linear distortions are corrected via a series of deep learning-based preprocessing techniques to minimize the domain gap between contact and corresponding contactless fingerprint images. In chapter 4, we aim to improve the sensor-interoperability of fingerprint recognition by leveraging a diversity of deep learning representations, integrating convolutional neural network and attention-

based vision transformer architectures into a single, multimodel embedding. Similarly, in chapter 5, we further improve the robustness and universality of fingerprint representations by fusing multiple local and global embeddings and demonstrate a marked improvement in latent to rolled fingerprint recognition performance, both in terms of accuracy and efficiency. Next, chapter 6 presents a method for synthetic fingerprint generation, capable of mimicking the distribution of real (i.e., bona fide) and PA (i.e., spoof) fingerprint images, to alleviate the lack of publicly available data for building robust fingerprint presentation attack detection algorithms. Finally, in chapter 7 we extend our fingerprint generation capabilities toward generating universal fingerprints of any fingerprint class, acquisition type, sensor domain, and quality, all to improve fingerprint recognition training and generalization performance across diverse scenarios.

To my loving friends and family.

ACKNOWLEDGEMENTS

As I reflect on my time as a PhD student, I am filled with tremendous joy from the experience as well as with an immense gratitude for the friends, colleagues, and mentors who have guided and instructed me along the way. Although there are too many to mention, I would like to give my thanks and appreciation for a few individuals who have truly impacted me during this time. Foremost, my deepest gratitude to my advisor, Professor Anil K. Jain, for his unwavering support and encouragement in both my personal and professional life. Thank you, Dr. Jain, for your mentorship and dedication to teaching me how to become an independent and successful researcher.

I would also like to thank my PhD committee, Dr. Arun Ross, Dr. Xiaoming Liu, and Dr. Kai Cao, for providing valuable feedback on this dissertation and for their collaboration throughout my PhD program. Thank you, Brenda Hodge, Amy King, and Vincent Mattison, for administrative and everyday assistance throughout the degree program. A special thank you to Russell Werner for not only all the impressive systems and computing support over the years which have saved me countless hours but also for the comical exchanges which made dealing with environment issues that much more bearable.

I am grateful for all my fellow labmates with whom I have shared many great memories over the years. A special thank you to Joshua Engelsma who has been an incredible mentor and friend. Thank you also to Tarang Chugh, Debayan Deb, Sixue Gong, Yichun Shi, Vishesh Mistry, and Divyansh Aggarwal with whom I had the pleasure of working with. Thank you to Kanishka Wijewardena, Akash Godbole, and Xiao Guo for sharing many fond memories together in the lab.

Thank you to all my loving family and friends who have always supported me. I am especially thankful to my parents for how they helped nurture the abilities and fortitude that God gave me to pursue my dreams. Thank you to my sister whose encouragement was much needed at times. Finally, thank you to my loving girlfriend, Allison Pasek, who has made the last few years that much more memorable. There are, of course, many other friends and family who have impacted me along the way, and I thank all of you.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The term “fingerprint recognition" evokes diverse mental images among individuals, ranging from cinematic spy scenarios featuring characters like James Bond to depictions of forensic leads in criminal investigative television series. In any case, fingerprint recognition has established itself as one of the most widespread and valuable methods of personal identification, culminating in centuries worth of scientific and technological discovery. With its popularity, fingerprint recognition has permeated into numerous areas of everyday life, including secure facility access, smartphone unlock, forensics, border control, and national identity programs [160]. On the surface, the ease of which many of us use fingerprint recognition in our daily lives can overshadow the technological advancements that underpin its development as well as overlook the need for future innovations to come.

In this chapter, we first explore the origins of fingerprint recognition and how it became so pervasive in modern society. After detailing its long history, we discuss some notable applications of fingerprint recognition in the real world that have aided the familiarity that most of us feel with the technology. Then, we introduce the basic components of an automated fingerprint recognition system before turning our attention to some of the potential pitfalls and challenges remaining with the technology. Finally, we conclude with a description of the specific solutions proposed in this dissertation to address many of those challenges.

1.1 History of Fingerprint Recognition

Human fascination with fingerprints dates back to 7000 BC, where the first discovery of thumbprints was found in Neolithic bricks from the ancient city of Jericho, in the State of Palestine [127]. Fingerprints were also found embedded into clay tablets in Babylon from the years 1955-1913 BC, suspected of being used in business contracts. Similarly, according the Chinese historian, Kia Kung-Yen, fingerprints were used from 600-700 AD during the Tang dynasty to sign legal documents [105]. These accounts speak to some degree of early belief in fingerprints as a means of personal identification; however, it was not until a century later that scientific studies on

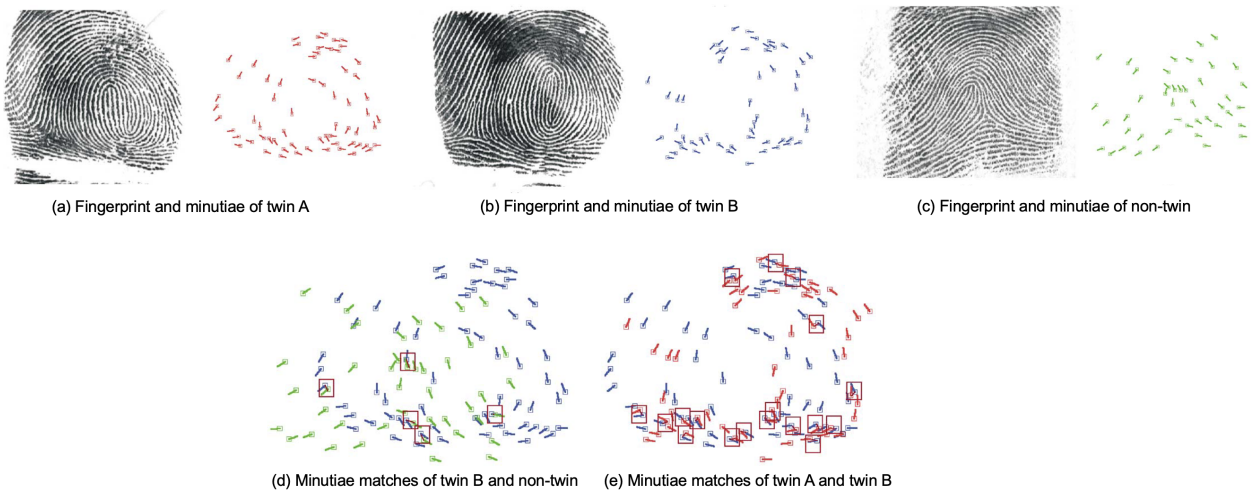


Figure 1.1 Example fingerprint images and corresponding minutiae from two identical twins, (a) twin A and (b) twin B, and one non-twin (shown in c) shown for comparison. Minutiae matching for (d) twinB–non-twin (matching of (b) and (c), matching score = 3 on a scale of 0–999) and (e) twin A–twin B (matching of (a) and (b), matching score = 38 on a scale of 0–999). The "matched" minutiae pairs are shown by bounding boxes. Though the match score of 38 between the twins is larger than the match score of 3 between twin B and the non-twin, this match score is still below the threshold for a genuine match. Figure reproduced from [122].

the merits of fingerprint recognition began. As we will see, these studies would come to establish two core tenets of fingerprint recognition, namely uniqueness and permanence:

1. Uniqueness: Due to genetics and random forces in play during the formation of friction ridge details, no two fingers, even for the same individual, have identical fingerprints. Even twins are reported to have unique fingerprints [122], as illustrated in Figure 1.1.
2. Permanence: Friction ridge patterns are believed to be persistent during the lifetime of an individual in terms of their ability for personal identification. For example, the fingerprint images shown in Figure 1.2 are from the same finger captured over a 12 year timespan. Despite small variations due to the introduction of small cuts and bruises, the ridge structure remained constant throughout the years of collection.

The uniqueness tenet of fingerprints came to fruition through multiple studies on the formation of ridges, furrows, and pore structures of fingerprints, which began in 1684 by English plant morphologist Nehemiah Grew [52]. A century later, in 1788, a detailed description of the anatomical formations of fingerprints was made by Mayer [174]. In 1892, Sir Francis Galton, a polymath and

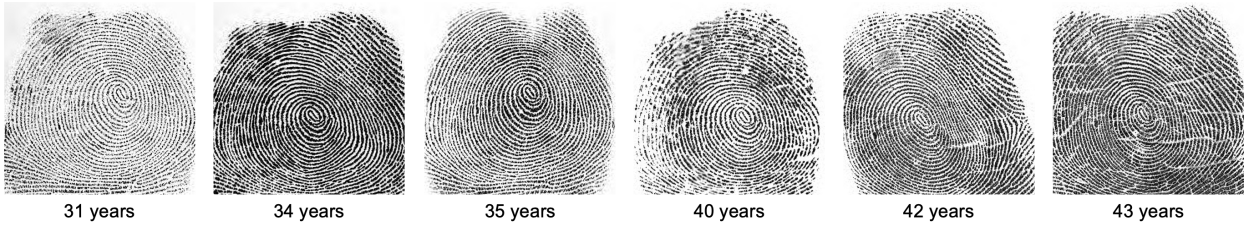


Figure 1.2 Time-lapse of the same finger over a 12 year period from the longitudinal dataset used in [262].

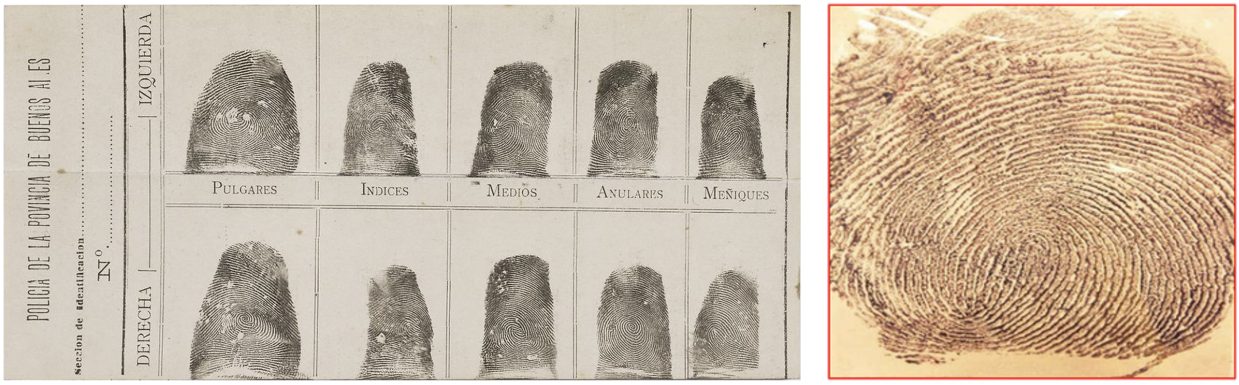
cousin of Charles Darwin, wrote the landmark book *Finger Prints* in which he stated the following on the potential of fingerprint ridges:

> "Let no one despise the ridges on account of their smallness, for they are in some respects the most important of all anthropological data. We shall see that they form patterns, considerable in size and of a curious variety of shape, whose boundaries can be firmly outlined, and which are little worlds in themselves. They have the unique merit of retaining all their peculiarities unchanged throughout life, and afford in consequence an incomparably surer criterion of identity than any other bodily feature." [81].

Going further, Galton made note of the uniqueness of fingerprint minutiae, the various endings and bifurcations found throughout the fingerprint ridge structure, which are still one of the most common feature representation used in automated fingerprint identification systems today [160].

On the other hand, the notion of permanence was first established by William Herschel, a German-born British astronomer, who demonstrated the permanence of fingerprints in his 1916 book titled *The Origin of Finger-Printing* [109], where he collected longitudinal inked impressions of his son's finger at the ages of 7, 17, and 40 years old and concluded that fingerprints remained constant over time. Furthermore, Dr. Henry Faulds observed that not only were fingerprints permanent, but they also grew back into the exact same pattern when the outer skin of the fingerprint was removed [52]. In other words, fingerprints were a permanent physical characteristic of an individual that remained with them throughout their lifetime.

Based on these two long-standing beliefs, fingerprint recognition began to gain prominence as a critical means of identification in law enforcement applications. In 1892, fingerprints were



(a) Rolled ten-print fingerprint images of Francisca Rojas

(b) Crime scene print found at the scene

Figure 1.3 (a) Rolled ten-print fingerprint images of Francisca Rojas and (b) matching crime scene fingerprint found on the door post at the scene of the murder of her two children in Buenos Aires, Argentina. Images obtained from [208] and [105].

used for the first time to solve a murder case of two children in Argentina [105]. Their mother, Francisca Rojas, confessed to the crime after being confronted by evidence matching one of her ten-print fingerprint images to a bloody fingerprint found on a door post at the scene (see Figure 1.3). In 1898, in the Bengal province of India, another murder case was solved using the remnants of two fingerprint impressions found on an almanac. Sir Edward Henry, Herschel's successor in India, found the prints matched with an ex-convict Kangali Charan whose thumbprint was already in the records due to a prior theft conviction [105]. In 1900, Sir Edward Henry introduced a scientific fingerprint classification system [108], which was later popularly known as Henry System of Classification. Henry's classification system consisted of five major fingerprint classes, which are demonstrated in Figure 1.4. It was officially introduced at New Scotland Yard for criminal identification in 1901 [105]. A major development happened in the year 1924 when the United States Congress mandated the collection of fingerprints of criminals. Consequently, a new identification division was instituted within the Federal Bureau of Investigation (FBI). In 1933, a unit specializing in technical analysis of latent fingerprints, i.e., noisy finger marks unintentionally left at a crime scene (such as the marks left on the almanac by Kangali Charan), was also established by the FBI [181].

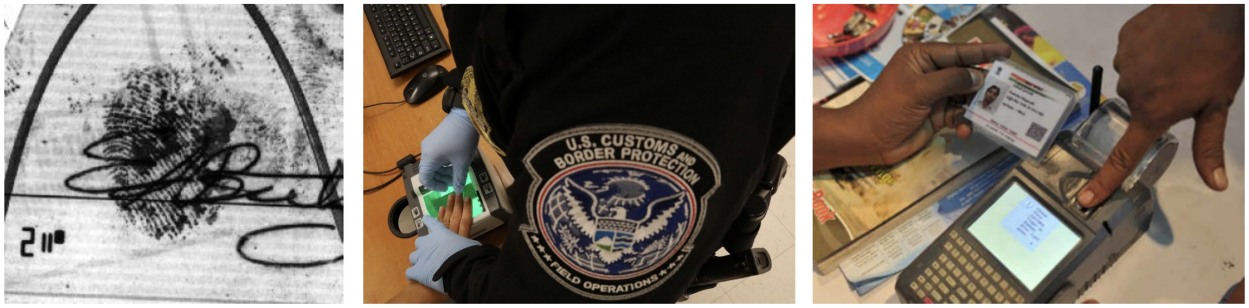
With the increasing use of fingerprints in law enforcement around the world, a need arose

Figure 1.4 Example images from the 5 major fingerprint categories established by Henry.

to automate the process of searching a fingerprint against the growing repository of enrolled fingerprints. This need was further supported by a report compiled by the RAND Corporation [91], which showcased the advantages and opportunities of the effective use of fingerprints as physical evidence in improving crime-solving performance. Recognizing the potential for the technology, combined with an electronics revolution happening at the time, agencies including the FBI, the UK Home Office, and the Japanese and French police agencies undertook research initiatives that led to development of Automated Fingerprint Identification Systems (AFIS) [133]. Consequently, the first algorithmic approach to fingerprint recognition was published in *Nature* in 1963 and the first Automated Fingerprint Identification Systems (AFIS) became a reality in 1974 [237].[1]

As accuracy and efficiency of AFIS improved, so did the scientific understanding of fingerprints thanks to rigorous statistical studies on central tenets of fingerprint recognition over the last few decades. In particular, it was long assumed fingerprints were unique and permanent, but this belief was only backed up scientifically in 2002 when Pankanti, Prabhakar, and Jain published "On the individuality of fingerprints" [190]. The authors computed the likelihood of two fingerprints sharing the exact same patterns to be approxiamtely $5.47x10^{-59}$, an incredibly small possibility. Similarly, in 2015, Yoon and Jain published a study in the *Proceedings of the National Academy of Sciences (PNAS)* providing strong backing to the permanence of fingerprints [262]. The study involved 15,597 subjects over time lapses of 5 to 12 years and found that, although match scores drop slightly over time, the overall recognition accuracy remained stable. These studies have helped solidify fingerprint recognition as a viable and secure means of person authentication, which has

[1]https://www.secureidnews.com/news-item/a-history-of-afis/



(a) Forensics (b) Border Crossings (c) National ID

(d) Consumer Products (e) Payments (f) Access Control

Figure 1.5 Example prominent applications of fingerprint recognition.

now permeated into myriad commercial and governmental applications. A number of these notable, well-known applications are discussed in the following section.

1.2 Major Applications

Due to the exceptional performance in terms of accuracy and speed exhibited by contemporary AFIS, centuries of empirical observation, and robust statistical evidence of the merits on which it is founded, fingerprint recognition has proliferated across a diverse range of applications in our present-day world. The following list highlights some of the more notable and widely recognized applications. Examples of these applications are illustrated in Figure 1.5.

- Forensics - In the mid to late nineteenth century, pioneers, such as Faulds, Herschel, and Henry, engaged in manual fingerprint examinations to identify repeat criminals [52]. In 1924, the FBI formally established the Identification Division to collect and store inked ten-print cards from criminals. By 1999, this process evolved into the FBI's Integrated Automated Fingerprint Identification System (IAFIS), where fingerprints (ten-prints) were digitized, stored, and automatically compared [133]. In 2011, the FBI introduced the Next Generation

Identification (NGI) system to enhance the outdated IAFIS, enabling faster and more accurate fingerprint recognition capabilities. Today, approximately 22,294 federal, state, local, tribal, and international partners submit criminal and/or civil electronic entries to this system per month, according to statistics gathered at the end of December 2023 [180].

- Border Crossings - The Office of Biometric Identity Management (OBIM) oversees the largest biometric repository in the United States. A key aspect of OBIM's purpose is in preventing criminals and dangerous individuals from entering the United States. The system employed, called the Automated Biometric Identification System or IDENT, currently holds approximately 300 million unique identities and processes more than 400,000 biometric transactions per day [179].
- National ID - India's Aadhaar program currently stands as the largest biometric recognition system in the world [1]. Aadhaar employs all 10 fingerprints, 2 irises, and a face image of each Indian citizen enrolled. One of the main objectives is to eliminate duplicates and associate each individual with a unique 12-digit identifier. It has proven to be an effective tool in facilitating financial transactions and in curbing instances of fraud that may occur when providing benefits to the marginalized population. As of November 7th, 2023, Aadhaar has enrolled over 1.3 billion Indian citizens [223].
- Consumer Products (e.g., laptops, smartphones, etc.) - According to a recent study, 81% of smartphones employ some kind of biometric authentication [45], with face and fingerprint among the most popular biometrics being used for this purpose. However, consumers report that the most acceptable biometric technology is fingerprint recognition, with 86% of Americans feeling comfortable with the technology, compared to 30% who express concerns about the use of facial recognition.[2]
- Payments - Major payment companies like Visa and Master Card are incorporating fingerprint recognition directly into credit cards through a concept known as "Match on Card" [189]. This concept allows users to enroll their fingerprint template onto a chip embedded in their

[2]https://passport-photo.online/blog/biometric-statistics

credit card, enabling them to conduct financial transactions without the need for a PIN number. Importantly, the fingerprint template data remains securely within the credit card, ensuring its confidentiality, and is directly matched to the queried fingerprint on the chip during transactions.

- Access Control – Popularized in numerous spy movies and TV shows over the last few decades (e.g., *James Bond* [100], *Mission Impossible* [99], etc.), fingerprint recognition is well recognized as a secure and reliable means of restricting access to sensitive and top-secret buildings and information. In addition to protecting sensitive information, commercial entities may also employ fingerprint recognition to restrict access to paying customers for events such as sporting events and concerts. One of the most well-known examples of this is perhaps Disney's use of fingerprint recognition to access their theme parks [50].

The aforementioned applications and statistics suggest that it is highly likely that the majority of the global population has personally encountered fingerprint recognition at some point in their lives. The data further indicates a potential for continued growth in these numbers. In the following section, we explore the technical aspects of the modern automated fingerprint recognition system and explore how these systems became so widespread in our everyday lives.

1.3 Pipeline of Automated Fingerprint Identification Systems

With advancements in fingerprint sensing technology and automated matching algorithms, fingerprint recognition has become highly accurate and efficient. A typical recognition system involves two key stages: enrollment and recognition. A diagram of these stages of a typical fingerprint recognition system is shown in Figure 1.6.

1. Enrollment: In this stage, an individual's fingerprint, acquired using a fingerprint reader, undergoes processing to extract salient features and generate a fingerprint template. This template is then tagged with a unique user identifier and stored with associated metadata in a database. This database is known as the reference, gallery, or enrollment database.
2. Recognition: Depending on the application context, the recognition of an individual can be for validating the claimed identity (verification) or establishing the identity of an unknown

individual (identification). In both cases, a fingerprint is acquired and processed to generate a template, known as the query or probe template.

a) Authentication: In the authentication scenario, also known as verification, the query template is accompanied by a user identifier (the claimed identity), which is used to retrieve the enrolled template from the reference database. The system accepts or rejects the submitted claim of identity by performing a one-to-one comparison between the query template and the retrieved reference template. Examples include fingerprint-based access control and large-scale civil ID systems (e.g., Aadhaar), where the user provides a unique ID (e.g., employee RFID card or Aadhaar 12-digit unique ID) and a fingerprint impression for authentication.

b) Identification: In the identification scenario, the system aims to establish the identity of a subject by searching the entire reference database for a match. Operating in the identification mode, a biometric system performs one-to-many comparisons to determine if the user is already enrolled in the database, returning the user identifier that matches. This scenario is common in criminal investigations, where a fingerprint left at the crime scene is used to search if the perpetrator is already enrolled in the database.

The functionality of each of the fingerprint recognition modes is facilitated by a configuration of multiple sub-processes, including fingerprint acquisition, feature extraction, and matching. Each of these sub-modules are described in detail below.

1.3.1 Fingerprint Acquisition

In the early stages of fingerprint recognition, fingerprint images were acquired via a process that involved inking a user's fingers and having them press down on card stock paper. The captured fingerprints could be either rolled fingerprints, obtained by rolling the finger from one side to another, or slap/plain fingerprints captured by pressing the fingers flat against the card. These fingerprints were then filed away and manually compared by an examiner. Over time, digital fingerprint readers were developed, offering significant convenience compared to the traditional

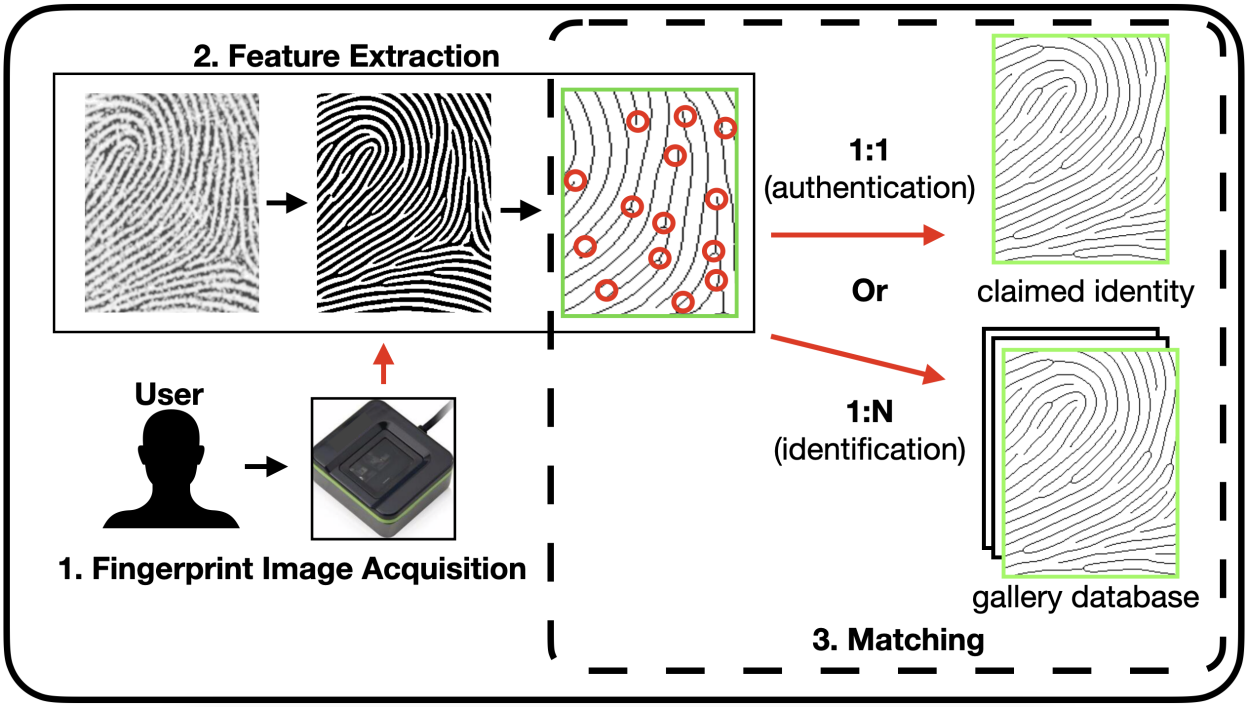


Figure 1.6 Diagram of typical fingerprint recognition pipeline.

"ink on paper" capture techniques. The various mechanisms for how these digital fingerprint readers captured images are discussed below.

1.3.1.1 Sensing Technologies

The major sensing technologies used in fingerprint readers over the years include, but are not limited to, ultrasound, frustrated total internal reflection (FTIR), direct-view imaging, capacitance, thermal, and pressure sensors. Depending on the imaging technology used, the acquired images may have starkly different characteristics and appearances, as illustrated in Figure 1.7. These differences can pose a significant challenge to fingerprint recognition algorithms designed to work on images acquired by one type of fingerprint reader when applied to images captured on a completely different device. This difficulty is enhanced if the two readers employ drastically different sensing technology. Chapter two of this dissertation will concentrate on enhancing the interoperability of fingerprint recognition systems across various sensor types. Further details on individual sensing mechanisms are outlined below:

1. Optical: Among optical sensing technologies, frustrated total internal reflection (FTIR) is

Figure 1.7 Example images from the NIST SD 302 dataset [78] of the same finger captured across 6 different fingerprint readers.

the most widely used. FTIR sensing involves using a light source and a glass prism, with a camera capturing reflected light from the fingerprint ridges and valleys, resulting in a high-contrast fingerprint image. Other optical fingerprint readers use direct-view imaging, where a light source illuminates the finger, and light from both ridges and valleys is reflected back towards the camera. Contactless optical readers typically offer lower contrast compared to FTIR systems but are more hygienic and less impacted by moisture on the finger due to environmental humidity.

2. Solid-State: Solid-state sensing technology operates using an array of mini-sensors measuring differentials in capacitance, temperature, or pressure between ridges and valleys of the finger. Due to their small size and low cost, solid-state sensors are commonly deployed in mobile devices [160].
3. Ultrasound: Ultrasound sensing emits acoustic waves toward the fingertip on the imaging platen and uses a receiver to gather the echoed response to develop a depth profile of the fingerprint. Ultrasound sensing is advantageous for capturing a subsurface fingerprint image which is useful for detecting fake fingerprint attacks and alleviating contaminants on the surface of fingerprints. Recently, QualComm Inc. developed an in-display ultrasound sensor for mobile phones, widely deployed in the Samsung smartphone series (Galaxy S10 onwards) [160].

With the onset of the COVID-19 pandemic, there has been a surge in interest toward contactless acquisition, especially utilizing direct-view cameras such as readily available smartphone cameras. However, for the field to move toward contactless acquisition, several significant challenges stem-

ming from the domain gap between images acquired via one of the aforementioned contact-based acquisition methods need to be addressed. In particular, the presence of non-linear distortion in contact prints introduces alignment errors between undistorted contactless prints. Paired with marked differences in the appearance of the images due to differences in lighting exposure, varying backgrounds, and varying stand-off between the finger and the imaging device, drastically different images may be captured between contactless and contact-based images of the same fingers. Varying perspective differences as the finger is unconstrained along 6 degrees of freedom as the image is being captured in space also presents another set of unique challenges, both for contactless to contactless matching and contact to contactless matching. Chapter three of the dissertation proposes several methods to reduce the domain gap between newly mobile phone acquired contactless fingerprint images and corresponding contact-based images that are commonly found in legacy databases.

1.3.2 Feature Extraction

To determine the identity of a given fingerprint image, a measurement (feature) space is required in which multiple fingerprint images of one identity (finger) are clustered together in the space and images belonging to another finger occupy a separate cluster in the space. Two main requirements of a good feature space are that the chosen features should be salient, meaning that the representation contains discriminative information about the fingerprint, and effective, meaning that the representation can be easily obtained, stored in a compact manner, and be useful for matching [160]. For purposes of recognition, a number of different fingerprint features have been proposed, which capture distinct information from the input fingerprint ridge structures. These features are commonly categorized in a hierarchical order based on the scale of the characteristics which they contain, as described below and visualized in Figure 1.8:

- Level-1: These global features encompass fingerprint pattern types (arch, loop, whorl), singular points (cores, deltas), ridge orientation, and ridge spacing. While useful for indexing and fingerprint alignment, they do not typically provide enough discriminative information for unique finger identification. Techniques such as image processing, detection of ridges with

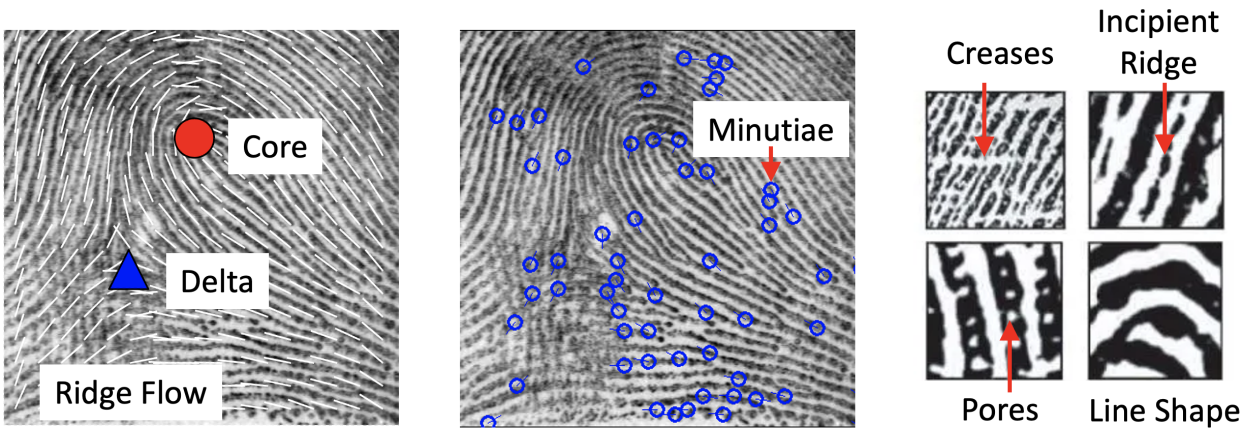


Figure 1.8 Illustration of the three scales of features typically extracted from a fingerprint image.

maximum curvature, or deep learning approaches are employed to extract these features [160].

- Level-2: These local features pertain to salient points where ridges exhibit discontinuity, such as ridge endings and bifurcations, known as minutiae points. There can be over 100 minutiae found in a rolled fingerprint, but the number of corresponding minutiae considered sufficient for a high-confidence match is believed to be around 12 to 15. A study from 2013 on the sufficiency of information for latent fingerprint examination cites 44 countries as having a minimum number of minutiae required for evidence, which can range from 4 to 16 points. For example, the United Kingdom requires a minimum sixteen points, while 24 other countries (e.g., Australia, Germany, etc.) only require twelve. The United States does not have a minimum [239].
- Level-3: These features involve fine-grained characteristics of fingerprints, such as sweat pores, incipient ridges, scars, creases, and dots between ridges. While providing additional uniqueness, they typically require a minimum scanning resolution of 1000 ppi for successful extraction. Mainly utilized by latent fingerprint examiners for manual comparison, these features are not commonly employed in AFIS due to their lack of robustness, cost of fingerprint reader construction, and slow extraction and matching speeds. However, recent developments in low-cost, high-resolution readers have led to the development of algorithms utilizing level-3 features for matching [120].

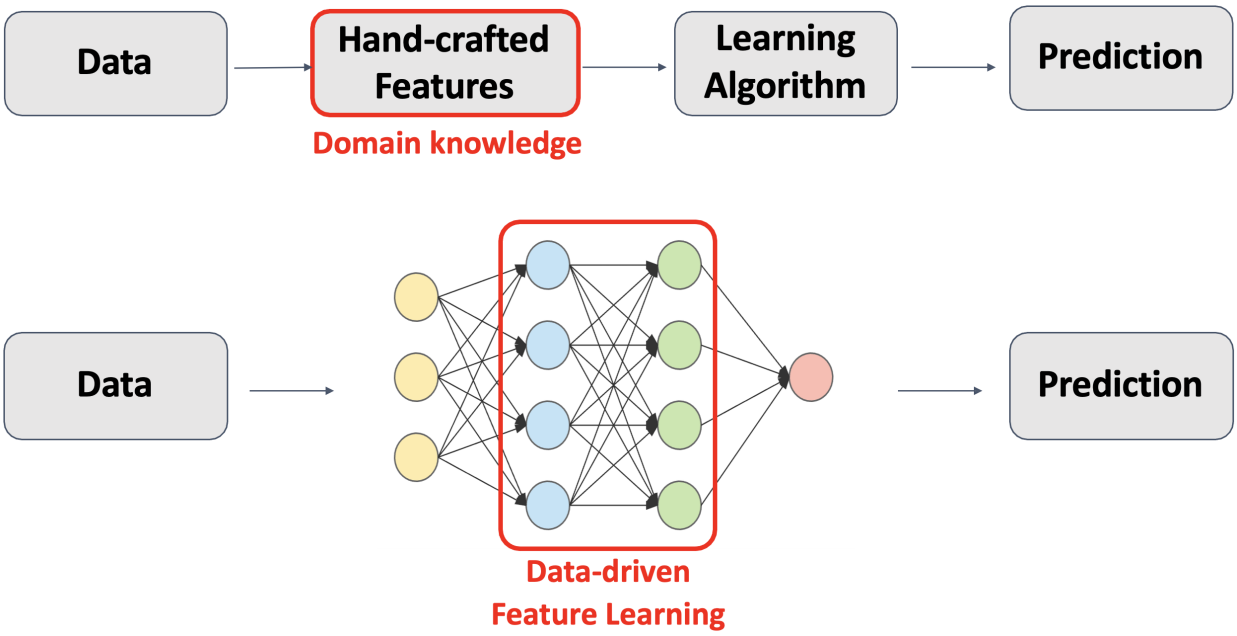


Figure 1.9 Handcrafted feature learning vs. data-driven feature learning.

State-of-the-art commercial-off-the-shelf matchers (COTS) may leverage either knowledge-driven (i.e., hand-crafted) techniques and/or data-driven methods, such as Convolutional Neural Networks (CNNs), for extracting fingerprint features, including those previously mentioned. Much of chapters 4 and 5 of this dissertation explores the complimentary nature of these representations, with an emphasis on how to successfully leverage a combination of them for improved accuracy and efficiency in fingerprint recognition. Each of these two paradigms, knowledge-driven and data-driven, are discussed further in the following sections within the context of fingerprint recognition (see also Figure 1.9 for an illustration of the differences).

1.3.2.1 Knowledge-driven Representation

The prominent characteristic of each fingerprint image is the organization of various connected ridges and valleys, which typically portray as dark pixels for ridges and bright pixels for valleys in the captured digital images [160]. Before the prevalence of big data and large-scale computing, designers of automated fingerprint recognition systems relied on extensive knowledge about the problem domain (fingerprints) to extract salient features from these ridges and valleys to discriminate between different fingerprint identities. Riding on the back of Galton's claims about the

uniqueness of minutiae points, many of these hand-crafted approaches were aimed at reliably extracting points of discontinuity along the ridges. The points were commonly denoted by their type (e.g., termination, bifurcation, etc.), x and y coordinates within the image, and the angle between the tangent to the connecting ridge line and the horizontal axis [160].

A number of preprocessing steps have proved useful for reliable minutiae extraction, including fingerprint image enhancement, segmentation, and binarization. Traditional algorithms for each of these steps employed the use of well-known image processing techniques, such as Gabor Filters for image enhancement [111], Prewitt and Sobel filters for ridge orientation estimation [87], sinusoidal-shaped waves for frequency estimation [154], and local histograms and morphological operations for fingerprint segmentation and binarization [170]. Finally, minutiae are detected from thinned versions of the binarized images by summing the differences between pairs of adjacent pixels in a neighborhood of each pixel [160].

1.3.2.2 Data-driven Representation

As advancements in deep learning proved remarkably useful across a wide array of problems in computer vision and machine learning, researchers began exploring the use of deep networks for various fingerprint recognition tasks. Some of these methods supplanted the hand-designed algorithms used for fingerprint feature extraction in the past; some examples include fingerprint enhancement [116, 126, 137, 271], segmentation [30], and minutiae extraction [29, 51, 187, 233]. From these developments arose a new approach to fingerprint representation in which a fingerprint image is input to a deep network (typically a CNN) and a single, fixed-length embedding is obtained. Among the first of the kind was a fingerprint representation network nicknamed DeepPrint, which would output a single 192-dimensional feature embedding [68]. What set DeepPrint apart from other fixed-length representation networks was its incorporation of minutiae features directly into the learning process. Thus, DeepPrint was able to show remarkable improvement in fingerprint recognition accuracy by embedding minutiae domain knowledge into the data-driven learning process of CNNs, while at the same time, delivering significant speedup in matching. A major focus of chapter 4 of this dissertation also aims to combine domain knowledge with data-driven

representations for improved performance of latent fingerprint recognition.

1.3.3 Matching

A fingerprint matching algorithm compares two given fingerprint templates and typically outputs a similarity score, usually a value between 0 and 1. A score close to 0 implies low similarity, while a value near 1 indicates very high similarity. A match score above a specified threshold (t) is considered a successful match. A strict threshold (close to 1) enhances security by minimizing false accepts but may lead to a poor user experience due to increased false rejects. Fingerprint matching poses a challenge due to significant variability between different impressions of the same finger (intra-class variability), influenced by factors such as noise, displacement, partial overlap, pressure, and skin conditions [160]. There are essentially three broad categories of fingerprint matching approaches:

1. Correlation-based matching: This technique involves overlaying two fingerprint images and calculating the correlation between corresponding pixels for different alignments, rotations, and displacements. Due to its resource-intensive nature, these techniques are not widely used.
2. Minutiae-based matching: This is the most popular and widely deployed technique for fingerprint matching, used by both automated algorithms and fingerprint examiners. It entails finding the alignment between the reference minutiae set and the input query minutiae set that results in the maximum number of paired minutiae.
3. Non-minutiae feature-based matching: In cases of low-quality images, such as latent fingerprints, minutiae extraction becomes challenging. This category of matching approaches may utilize ridge pattern characteristics (e.g., local ridge frequency and orientation) or texture information using hand-crafted or deep learning methods [31]. Combining minutiae-based and texture-based features can significantly enhance the matching performance of latent fingerprints, including state-of-the-art deep-learning based methods with fixed length representation [68].

Besides accuracy, another important component of fingerprint matching algorithms is the

latency or time taken to return a match. For 1:1 authentication systems, the latency time becomes a nuisance factor for patrons utilizing the system, but anything close to real-time is sufficient to achieve a satisfactory experience. For search applications (1:N comparison), however, the latency can have a significant impact on the usability of the system with a large number of comparisons (e.g., tens of millions). For example, with the FBI's NGI system, which maintains a database of roughly 185 million fingerprints of criminal, civil, and military individuals, the match speed becomes a significant [101] factor when searching a latent fingerprint probe to find a potential suspect match. Chapter 5 of this dissertation discusses a mutli-stage search strategy to significantly reduce the time required to search a gallery given an input latent fingerprint probe.

1.4 Challenges in the Generalization of Fingerprint Representations

Despite notable progress in the evolution and acceptance of fingerprint recognition over the past few decades, substantial challenges persist and require attention. As mentioned in the introduction, a significant portion of these challenges arises from the fragility of fingerprint feature extraction and matching in the face of noisy, low-quality prints, diverse capture characteristics, and the scarcity of data for acquiring robust and broadly applicable features. The ensuing sections outline some of these specific challenges which this dissertation aims to address.

1.4.1 Sensor-interoperability: Lack of Sensor-invariant Features

A well known issue of data-driven machine learning approaches is their susceptibility to overfit to the training data [60]. The general idea of overfitting in machine learning is when a model memorizes the training data too well, including its noise and specificities, to the extent that it negatively impacts the model's generalization performance on new, unseen data. Given the high variability in the appearance of fingerprint images captured via various sensing mechanisms, learning robust, sensor-invariant features is of particular importance and interest in the fingerprint recognition community.

1.4.1.1 Universal Fingerprint Representation

As introduced in Section 1.3.2, many different types of features have been proposed for fingerprint recognition within the last century, which can be broadly classified into one of two categories: knowledge-based and data-driven. The most popular knowledge-based feature is that of minutiae points, which have become an essential component of automated fingerprint recognition systems. More recently, deep learning-based fingerprint representations (i.e., embeddings) have emerged as a viable and complementary alternative to minutiae features. However, minutiae features and deep learning-based embeddings each have their own advantages and disadvantages. Minutiae, which are explainable and have a strong background in scientific understanding, perform exceptionally well in high-quality fingerprint images, particularly in large area fingerprints which may contain a large number of minutiae, but their performance degrades rapidly in partial and very noisy fingerprint images, such as latent fingerprints or individuals with degraded finger skin [98]. On the other hand, fixed-length embeddings seem to improve the performance in low-quality and partial fingerprints compared to minutiae, and are orders of magnitude faster to match [68]. However, they suffer from a lack of transparency and explainability. Research into improved methods on how to leverage both representations for robust, universal fingerprint recognition is an area of active research.

1.4.1.2 Cross-sensor and Cross-material Spoof Detection

One of the most significant challenges threatening the security of fingerprint recognition devices today is that of presentation attacks from adversaries trying to gain unauthorized access to these systems [165, 178]. A presentation attack (PA), as defined by the ISO standard *IEC 30107-1:2016(E)*, is a "presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system." [117]. Common presentation attack artifacts include fingerprint casts constructed from molds using readily available household materials (gelatin, silicone, wood glue, etc.) that aim to mimic the ridge-valley structure of an enrolled user's fingerprint [25, 66, 162, 169, 261]. This concern has led to a series of competitions on fingerprint presentation attack detection (PAD) methods to alleviate the vulnerability of these systems to presentation attacks.

(a) (b)

Figure 1.10 Example raw and segmented (a) contactless and (b) latent fingerprints.

As of recently, convolutional neural network (CNN) approaches have shown the best performance on the respective genuine vs. PA benchmark datasets. However, due to the different optical and mechanical properties, it has been shown that the PAD error rates of these approaches suffer up to a three-fold increase when applied to datasets containing PA materials not seen during training, denoted as “cross-material generalization" [163, 229]. Similarly, a performance gap exists for “cross-sensor generalization", in which presentation attack algorithms are applied to fingerprint images captured on new fingerprint sensor devices that were not seen during training.

1.4.2 Unconstrained Fingerprint Recognition

Two notable applications of fingerprint recognition in the last few years are in the unconstrained capture scenarios of contactless fingerprints, mostly by mobile phones which obviates the need for a separate fingerprint reader, and latent fingerprints left unintentionally at crime scenes. Contactless acquisition of fingerprint images suffers from a high degree of variability due to the six degrees of freedom during capture and latent prints are often extremely smudgy, occluded, and distorted (see Figure 1.10 for examples). The following two sections will discuss each of these challenging capture scenarios in detail.

1.4.3 Contact to Contactless Fingerprint Matching Compatability

Despite the benefits of contactless fingerprint acquisition, imaging and subsequently matching a contactless fingerprint presents its own set of unique challenges. These include (i) low ridge-valley contrast, (ii) non-uniform illumination, (iii) varying roll, pitch, and yaw of the finger, (iv) varying background, (v) and perspective distortions due to the varying distances of the finger from the camera. Of particular importance is the lack of cross-compatibility with legacy databases of

contact-based fingerprints. Indeed, for widespread adoption of contactless fingerprint acquisition, specific algorithms and methods need to be developed to minimize the domain gap between contactless fingerprints and corresponding contact-based impressions.

1.4.4 Latent Fingerprint Recognition

The reliability of automatic latent to rolled fingerprint matching considerably lags that of rolled to rolled fingerprint matching. As a result, some innocent individuals, like in the case of Brandon Mayfield [182], have unfortunately been incarcerated due to inaccurate latent to rolled comparison by automatic fingerprint identification systems (AFIS) and failure of forensic examiners to follow the FBI's ACE-V protocol established in the 1980s [8]. Some of the reasons for low performance in latent fingerprint recognition include poor ridge-valley contrast, occlusion, distortion, varying background, and incomplete fingerprint patterns. Because of these challenges, latent fingerprint recognition remains one of the most challenging problems in biometrics, akin to matching poor quality face images from CCTV surveillance frames to mugshot photos.

Few studies have focused on an end-to-end system to improve the latent to rolled fingerprint recognition pipeline, which is necessary since optimizing individual components separately may lead to sub-optimal performance when integrated together and tested as a complete system. Of those studies that do report on an end-to-end recognition system [28, 31, 234], the highest rank-1 retrieval rate reported in the academic literature is 65.7% [31], computed on 258 latent probes from NIST SD 27 against a background of 100K rolled fingerprints. However, increased interest in latent fingerprint recognition has led to the creation of the on-going Evaluation of Latent Fingerprint Technologies (ELFT) competition held by the National Institute of Standards and Technology (NIST), where these results are contiually improving [2].

1.4.5 Limited Public Domain Datasets

Recently, the community has seen a push toward deep neural network (DNN) based models for fingerprint recognition [26,68,93,139,143,217,219]. These compact, fixed-length embeddings can be matched efficiently and combined with homomorphic encryption for added security [74]. Indeed, this push toward DNN-based fingerprint recognition comes in the wake of the success demonstrated

in the face recognition domain in applying DNN models to face recognition, which was aided by the availability of large-scale face recognition databases scraped from the web, despite the many ethical and privacy concerns which have led to many of these datasets to be recalled today. Arguably, at least in part, the reason for the delayed adoption of DNNs for fingerprint recognition has been the lack of publicly available, large-scale fingerprint recognition datasets and increased scrutiny over privacy of biometric data, which has led to many works to generate synthetic fingerprint images [6, 9, 11, 19, 21, 23, 36, 71, 76, 125, 171, 173, 198, 252, 268].

Similarly, there has been an increased interest in DNN-based models for fingerprint presentation attack detection (PAD), i.e., spoof detection, where the scale and amount of publicly available data is also limited. Compounding the problem is the difficulty in collecting large-scale fingerprint PA datasets due to the increased time and complexity in fabricating and imaging artifacts mimicking realistic fingerprint ridge-valley structures. Further advancements in synthetic fingerprint generation are needed to improve the performance of models in the presence of a limited amount of real fingerprint data.

1.5 Thesis Contributions

This dissertation aims to improve the generalization performance of fingerprint recognition and fingerprint spoof detection in each of the challenging applications previously mentioned.

- **Sensor and Material Agnostic Fingerprint Presentation Attack Detection**

  To improve the generalization of fingerprint presentation attack detection across novel PA materials and fingerprint sensing devices, we propose an approach which builds off any existing CNN-based architecture trained for fingerprint liveliness detection. First, we incorporate a style transfer network wrapper to augment the training data with additional domain characteristics. Next, we utilize adversarial representation learning (ARL) to learn sensor and material invariant representations. These two approaches combined were shown to improve the cross-sensor (cross-sensor and cross-material) generalization performance from a TDR of 88.36% (78.76%) to a TDR of 93.03% (88.49%) at a FDR of 0.2%.

- **Contact to Contactless Fingerprint Matching**

We present a comprehensive, end-to-end solution for contact-contactless fingerprint matching that addresses the challenges inherent to each step in the contact to contactless matching process (mobile capture, segmentation, enhancement, scaling, non-linear warping, representation extraction, and matching). Our approach utilizes several deep learning-based pre-processing techniques to minimize domain gap between contact and contactless fingerprints. Additionally, a new dataset of 9,888 2D contactless and corresponding contact-based fingerprint images from 206 subjects (2 thumbs and 2 index fingers per subject) was collected as part of this work and made public to advance much needed research in this area. The smartphone contactless fingerprint capture app that was used to collect the dataset was made available as well.

- **Universal Fingerprint Representation via Multi-model Embeddings**

  To address difficult edge-cases where accurate fingerprint recognition remains challenging, such as partial overlap between two candidate fingerprint images and cross-sensor interoperability (e.g., optical to capacitive, contact to contactless, latent to rolled fingerprints, etc.), we introduce a novel deep learning-based fingerprint recognition architecture, AFR-Net (Attention-Driven Fingerprint Recognition Network), consisting of a shared feature extraction and parallel CNN and attention classification layers. This dual CNN and Vision Transformer (ViT) network successfully leverages the complimentary representations of CNN and attention-based networks for improved recognition accuracy across several diverse fingerprint datasets and sensor domains. Furthermore, we proposed a two-stage matching algorithm to significantly improve recognition performance in presence of low-quality and/or partial fingerprints, which was motivated from the observation the intermediate feature maps of deep learning-based fingerprint recognition networks encode local features that are also useful for relating two candidate fingerprint images. We use these corresponding local features between two candidate fingerprint image pairs to guide the network in placing attention on overlapping regions of the images. This leads to a more accurate determination of whether the images are from the same finger, especially for fingerprint pairs whose similiarities are

close to the match threshold.

- **Latent Fingerprint Recognition: Fusion of Local and Global Embeddings**

  To address the degradations commonly found in latent fingerprints (poor ridge-valley contrast, occlusion, distortion, varying background, and incomplete fingerprint patterns, etc.), we describe a novel, deep learning-based latent fingerprint enhancement method, coupled with an end-to-end pipeline for latent fingerprint recognition which leverages both a learned, global fingerprint representation (i.e., entire friction ridge pattern) and local representations (i.e., minutiae and virtual minutiae[3]) for improved accuracy and search speed of latent to rolled fingerprint recognition. Unlike existing latent to rolled fingerprint matching pipelines which are highly tuned specifically for latent fingerprints, our learned representation and matching pipeline is generalizable and effective across a wide range of fingerprint sensors (e.g., optical, capacitive, etc.) and image domains (e.g., latent, rolled, plain, contactless captures via mobile phone cameras, etc.).

- **Synthetic Fingerprint Spoof Images**

  To address the lack of large-scale fingerprint PA datasets, this dissertation describes a synthetic fingerprint generator, SpoofGAN, which is capable of generating highly realistic fingerprint impressions of both real (i.e., bona fide) and spoof (i.e., presentation attack) fingerprint images. SpoofGAN is a multi-stage generative architecture to fingerprint generation which can generate multiple, realistic fingerprint impressions from the same finger and a large number of different, unique fingers. We validate the realism of our synthetic bona fide and PA images through extensive qualitative and quantitative metrics including NFIQ2 [228], minutiae statistics, match scores from a SOTA fingerprint matcher, and T-SNE feature space analysis showing the similarity of real bona fide and PA embeddings to the embeddings of our synthetic bona fide and PA fingerprints. Besides verifying the realism of our synthetic PA generator, we also show how SpoofGAN fingerprints can be used to train a DNN for fingerprint PAD. We show this by improving the performance of a PAD model by augmenting

[3]Virtual minutiae are densely sampled points on an evenly spaced grid on the extracted fingerprint ridge area.

an existing fingerprint PA dataset with additional samples from our synthetic generator.

- **Universal Fingerprint Generation**

  Existing methods for generating fingerprints have limitations in creating varied impressions of the same finger with useful intra-class variations and diversity. To tackle this challenge, we present GenPrint, a framework designed to produce fingerprint images of various types while maintaining identity and offering humanly understandable control over different appearance factors such as fingerprint class, acquisition type, sensor device, and quality level. Unlike previous fingerprint generation approaches, GenPrint is not confined to replicating style characteristics from the training dataset alone: it enables the generation of novel sensor and style attributes from unseen fingerprint acquisition devices during inference without requiring additional fine-tuning. To accomplish these objectives, we developed GenPrint using latent diffusion models with multimodal conditions (text and image) for consistent generation of style and identity. Our experiments leverage a variety of publicly available datasets for training and evaluation. Results demonstrate the benefits of GenPrint in terms of identity preservation, explainable control, and universality of generated images. Importantly, the GenPrint-generated images yield comparable or even superior accuracy to models trained solely on real data and further enhances performance when augmenting the diversity of existing real fingerprint datasets.

CHAPTER 2

SENSOR AND MATERIAL AGNOSTIC FINGERPRINT PRESENTATION ATTACK DETECTION

This chapter addresses the poor generalization of existing presentation attack detection (PAD) solutions to new presentation attack (PA) materials and fingerprint sensors not seen during training. We present a robust PAD solution via a combination of adversarial representation learning (ARL) and a CNN-based architecture embedded with a cross-sensor style transfer network wrapper. The style transfer component aims to address the challenge of training robust deep networks on limited training data, whereas ARL is leveraged to encode generalized representations across all classes of sensors and PA materials. Experimental results on MSU-FPAD, Government Controlled Test (GCT), and LivDet 2015 and 2017 public domain datasets exhibit the effectiveness of the proposed approach in improving the cross-material and cross-sensor generalization performance.

2.1 Introduction

It has been observed that the security of automated fingerprint recognition systems may be compromised by presentation attacks from adversaries trying to gain unauthorized access to these systems [165, 178]. A presentation attack (PA) as defined by the ISO standard *IEC 30107-1:2016(E)* [117] is a "presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system." Common presentation attack artifacts include fingerprint casts constructed from molds using readily available household materials (gelatin, silicone, wood glue, etc) that aim to mimic the ridge-valley structure of an enrolled user's fingerprint [25, 66, 162, 169, 261]. Several examples of these artifacts are shown in Figure 2.1.

This concern has led to a series of competitions on fingerprint PAD methods to alleviate the vulnerability of these systems to various PAs. The First International Fingerprint Liveness Detection Competition debuted in 2009 [166] with subsequent competitions every two years, the most recent

This chapter was previously published as S. A. Grosz, T. Chugh, and A. K. Jain, "Fingerprint Presentation Attack Detection: A Sensor and Material Agnostic Approach", IEEE International Joint Conference on Biometrics, Houston, TX, Sept. 2020.
In the literature, presentation attack detection (PAD) is also commonly referred to as spoof detection and liveness detection. In this work, we use these terms interchangeably.

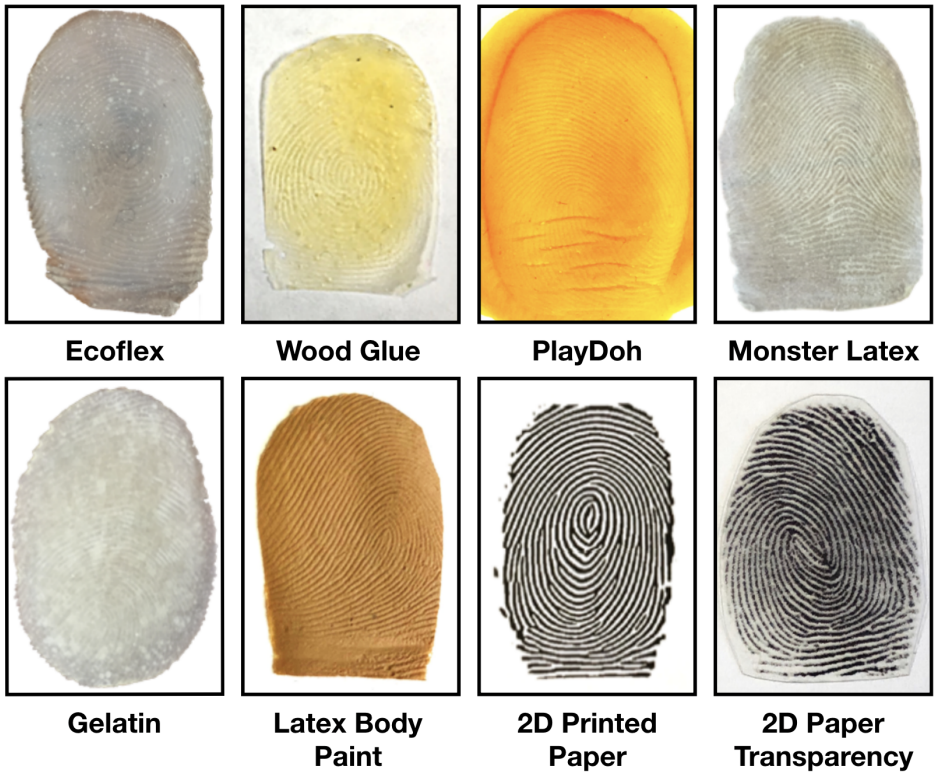


Figure 2.1 Example PA artifacts. Due to the varying visual and physical properties, many PAD algorithms fail to detect PA materials not seen during training.

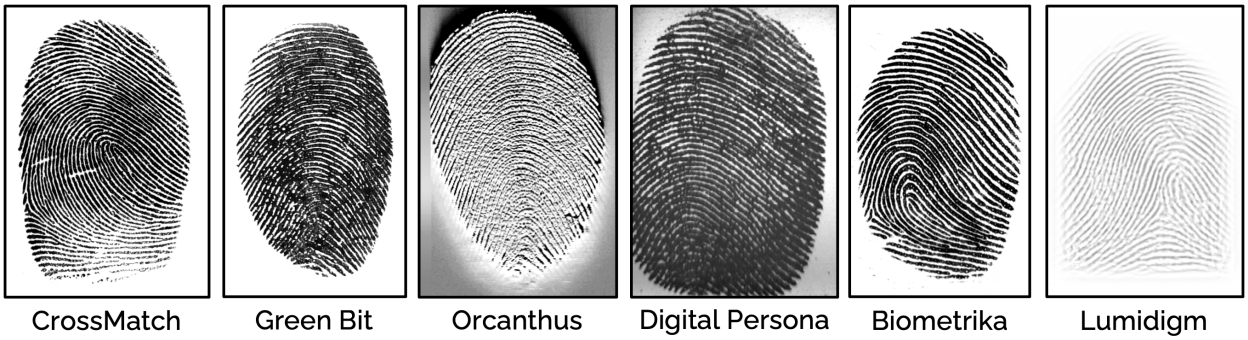


Figure 2.2 Illustration of the differences in textural appearance of live fingerprints captured on six different fingerprint readers. Images from LivDet 2015 [175], LivDet 2017 [176], and MSU-FPAD datasets [41].

being 2019 [86, 175, 176, 186, 257]. Approaches to fingerprint PAD can be broadly classified as either hardware-based, software-based or a combination of both. Hardware-based methods leverage the physical properties of various PA materials and use a number of additional sensors to gain further insight into the liveness of the presented fingerprint [12, 67, 135]. Some examples of sensing technologies that are inherently well-suited for liveness detection that have been used for

Table 2.1 Summary of published fingerprint cross-material generalization studies[1].

Study	Approach	Database	Performance
Rattani et al. [196]	Weibull-calibrated SVM	LivDet 2011	EER = 19.70 %
Ding & Ross [58]	Ensemble of multiple one-class SVMs	LivDet 2011	EER = 17.06 %
Chugh & Jain [41]	MobileNet-v1 trained on minutiae-centered local patches	LivDet 2011-2015	ACE = 1.48 % (LivDet 2015), 2.93 % (LivDet 2011, 2013)
Chugh & Jain [42]	Identify a representative set of spoof materials to cover the deep feature space	MSU-FPAD v2.0, 12 PA materials	TDR = 75.24 % @ FDR = 0.2 %
Engelsma & Jain [72]	Ensemble of generative adversarial networks (GANs)	Custom database of 12 PA materials	TDR = 49.80 % @ FDR = 0.2 %
Gonzalez-Soler et al. [88]	Feature encoding of dense-SIFT features	LivDet 2011-2015	TDR = 7.03 % @ FDR = 1.0 % (LivDet 2015), ACE = 1.01 % (LivDet 2011, 2013)
Tolosana et al. [235]	Fusion of two CNN architectures trained on SWIR images	Custom database of 8 PA materials	EER = 1.35 %
Gajawada et al. [79]	Style transfer from spoof to live images with a few samples of target material	LivDet 2015, CrossMatch sensor	TDR = 78.04 % @ FDR = 0.1 %
Chugh & Jain [44]	Style transfer between known spoof materials	MSU-FPAD v2.0, 12 PAs, LivDet 2017	TDR = 91.78 % @ FDR = 0.2 % (MSU-FPAD v2.0), TDR = 80.74 % @ FDR = 1.0 % (LivDet 2017)
Grosz et al. [92]	Style transfer with a few samples of target sensor fingerprint images + SARL	LivDet 2015, LivDet 2017, MSU-FPAD	TDR = 87.86 % @ FDR = 0.2 % (cross-sensor and cross-material LivDet 2015)
Proposed Approach	Style transfer with a few samples of target sensor fingerprint images + SARL + MARL	LivDet 2015, LivDet 2017, MSU-FPAD, GCT3	TDR = 88.49 % @ FDR = 0.2 % (cross-sensor and cross-material LivDet 2015)

[1] Several of the earlier studies used EER and ACE to characterize performance; however, in practice, TDR at a fixed FAR or the ROC curve is required.

fingerprint PAD are the multispectral Lumidigm sensor and OCT-based sensors [43]. In contrast, software-based solutions use only the information contained in the captured fingerprint image (or a sequence of images) to classify a fingerprint as bonafide or PA [41, 84, 85, 164, 167, 177, 188]. As of recently, convolutional neural network (CNN) approaches have shown the best performance on the respective genuine vs. PA benchmark datasets. However, due to the different optical and mechanical properties (see Figure 2.2), it has been shown that the PAD error rates of these approaches suffer up to a three fold increase when applied to datasets containing PA materials not seen during training, denoted as “cross-material generalization" [163, 229].

Some published studies aimed at reducing the performance gap due to cross material evaluations are summarized in Table 5.1. These approaches generally fall under one of two categories of

techniques: (i) single-class PAD [59, 73, 197] and (ii) synthetic data augmentation via style transfer or domain generalization [44, 79]. Many of these methods have shown incredible promise toward cross-material PAD generalization and have been applied toward other biometric modalities as well, including face [7] and iris [255]. The single-class methods circumvent the need to collect extensive training sets of a diverse class of PA materials, but they tend to lag the performance of binary-classifiers on known material attacks. On the other hand, synthetic data generators via domain generalization or style transfer are effective in augmenting the number of data samples of target materials of which we have only a few examples; however, they cannot approximate all possible data distributions of unknown attacks.

A similar performance gap exists for *cross-sensor generalization*, in which presentation attack algorithms are applied to fingerprint images captured on new fingerprint sensing devices that were not seen during training. One explanation for the challenge of cross-sensor generalization is the different textural characteristics in the fingerprint images from different sensors (see Figure 2.2). This discrepancy in the representation performance between the "seen source domain" and the "unseen target domain" has been referred to as the "domain gap" in the deep learning literature [16]. The cross-sensor evaluation can be considered as two separate cases: (i) all sensors in the evaluation employ the same sensing technology, e.g., all optical FTIR, and (ii) the sensors may vary in the underlying sensing mechanisms used, e.g., optical direct-view vs. capacitive.

In this work, we aim to improve the fingerprint presentation attack detection generalization across novel PA materials and fingerprint sensing devices. Our approach builds off any existing CNN-based architecture trained for fingerprint liveliness detection by encouraging cross-sensor generalization via a style transfer network wrapper. We also incorporate adversarial representation learning (ARL) in deep neural networks (DNN) to learn sensor and material invariant representations for presentation attack detection.

The main contributions of this chapter are enumerated below:

Generally, fingerprint sensor refers to the fingerprint sensing mechanism (e.g., camera and prism for FTIR optical, direct-view camera, thermal measurement device, etc.) and fingerprint reader refers to the entire process of converting a physical fingerprint into a digital image. For purposes of this study, we use these two terms interchangeably.

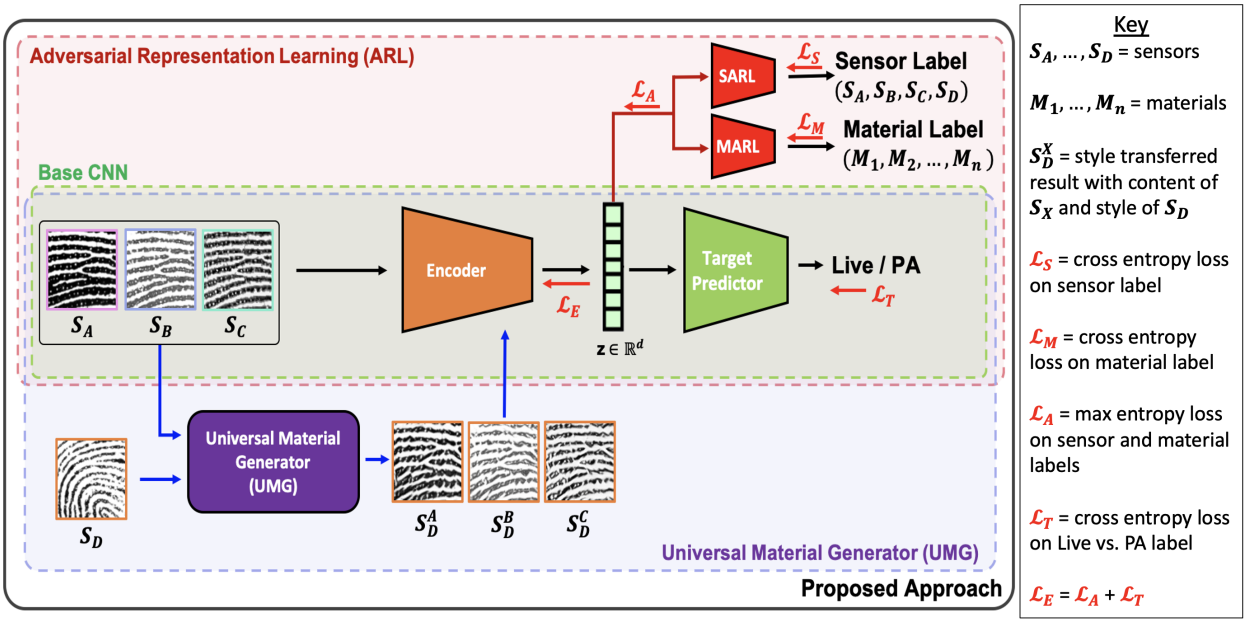


Figure 2.3 Overview of the network architecture for the proposed approach toward a more generalizable presentation attack detector.

1. A robust PAD solution with improved cross-material and cross-sensor generalization performance.
2. Our solution can be built on top of any CNN-based fingerprint PAD solution for cross-sensor and cross-material PA generalization using adversarial representational learning.
3. Experimental evaluation of the proposed approach on publicly available datasets LivDet 2015, LivDet 2017, MSU-FPAD, and GCT3. Our approach is shown to improve the cross-sensor (cross-sensor and cross-material) generalization performance from a TDR of 88.36% (78.76%) to a TDR of 93.03% (88.49%) at a FDR of 0.2%.
4. Feature space analysis of cross-sensor domain separation of the embedded representations prior to and following adversarial representation learning.
5. Detailed discussion of the challenges and techniques involved in applying deep-adversarial representation learning for fingerprint PAD.

2.2 Related Work

In this section, we review the previous approaches to cross-material generalization of fingerprint PAD, which generally fall under two broad categories: (i) single-class classifiers and (ii)

synthetic data augmentation. We also discuss the preliminaries of domain adaptation and domain generalization in the context of machine learning. Csurka provides a more in-depth review of domain adaptation [46]. Similarly, Wang and Deng provide a specific survey of the recent deep domain adaptation methods [244]. We also describe adversarial representation learning (ARL) as it is applied to the tasks of domain adaptation and domain generalization.

2.2.1 Single-Class Classification for PAD

Unlike cross-sensor PAD generalization, generalization to unseen PA species has received greater interest in the biometrics research community. One popular technique toward overcoming the challenge of unknown PAs is to design a one-class classifer that is trained to precisely learn the data distribution surrounding bonafide training examples and form a tight decision boundary around this distribution. In this way, any future test example which falls on the outside of the decision boundary will be classified as a presentation attack.

The advantage of the one-class approach is that the classification is independent of the location of any particular PA species in the high dimensional feature space, so long as it does not intersect with the space occupied by the bonafide representations. Preliminary works in fingerprint [59, 73, 197], face [7], and iris [255] PAD have shown incredible promise toward cross-material generalization; however, these techniques tend to lag performance of binary classifiers on seen materials.

2.2.2 Cross-Material PAD Generalization via Style Transfer

A separate approach to cross-material generalization leverages style transfer to synthesize additional training data to approximate unknown PA material representations. Gajawada et al. [79] first proposed data augmentation via a style transfer network wrapper to synthetically augment the amount of PA training data for a specific target PA material. This approach helps overcome the challenge of collecting sufficient training data amenable to training deep neural networks, which have shown superior PAD performance. Chugh et al. [44] extended this idea to interpolate between known PA materials to digitally synthesize new PAs, which improved the cross-material generalization. Style transfer is a very effective data augmentation tool to increase the robustness of the learned representations; however, the diversity of PA materials which can be synthesized

does not cover the full spectrum of possible PAs to be seen.

2.2.3 Domain Adaptation and Domain Generalization

A domain refers to a probability distribution over which data examples are drawn. In this context, domain adaptation and domain generalization are approaches to machine learning aimed at minimizing the performance gap between training data examples from a seen “source" domain and testing data from a related but different “target" domain. Therefore, domain adaptation and domain generalization are applied to situations in which the training and testing data points are not both independently and identically sampled from the same distribution. While domain adaptation involves training on labeled examples from the source domain and unlabeled data from the target domain, domain generalization assumes no access to labeled or unlabeled data examples from the target domain.

2.2.4 Adversarial Representation Learning (ARL)

Adversarial representation learning is a machine learning technique that can be applied to both domain adaptation and domain generalization. Adversarial representation learning has been applied in DNN architectures to extract discriminative representations for a given target prediction task (e.g., face recognition), while obfuscating some undesired attributes present in the data (e.g., gender information) [65, 82, 238, 263].

The general setup of ARL involves (i) an encoder network, (ii) a target prediction network, and (iii) an adversary network. The encoder network aims to extract a latent representation ($\mathbf{z}$) that is not only informative for the target prediction task (t), but also does not leak any information for the sensitive task (s). Meanwhile, the adversary network is tasked with extracting the sensitive information from the encoded latent representation. The entire network is trained in a minimax game similar to the generative adversarial networks introduced by Goodfellow et al. [89].

In Xie et al., the parameters of the adversary network are optimized to maximize the likelihood of the sensitive label prediction, whereas the encoder is trained to maximize the likelihood of the target task while minimizing the likelihood of the sensitive task [254]. In contrast, our proposed work is more aligned with the approach of Roy and Bodetti [206], where the adversary network is

optimized to maximize the likelihood of the sensitive label prediction from the latent representation, and the encoder is trained to maximize the entropy of the sensitive label prediction. In this manner, the base network is encouraged to encode a representation that aims to confuse the sensitive label prediction such that the adversary predicts equal probabilities (maximum entropy) for all classes of the sensitive label.

In tandem with our preliminary publication of this work combining style transfer with adversarial representation learning, Pereira et al. [193] also applied adversarial learning to encourage improved cross-material generalization. Our work stands out as we also incorporate adversarial learning on the sensor domains to learn a sensor-invariant as well as a material-invariant representation.

2.3 Proposed Approach

Our proposed approach toward fingerprint PAD generalization is a combination of style transfer and adversarial representational learning. For the style transfer component, we have adapted the Universal Material Generator (UMG) from Chugh et al. [44] to transfer style between sensor domains rather than between PA material types. Additionally, we include ARL to enforce invariance amongst the various sensing devices and unseen PA materials. An overview of the approach, which highlights each of the individual components, is shown in Figure 5.2. We discuss the workings and motivations of each individual component of the algorithm in the following sections.

2.3.1 Base CNN

What we refer to as the “base CNN" approach is a CNN trained on 96 x 96 aligned, minutiae-centered patches for classifying a given fingerprint impression as live or PA. It was shown by Chugh and Jain [41] that utilizing minutiae patches, as opposed to whole images, has two stark advantages: (i) it alleviates the difficulty of limited training data in the existing bonafide vs PA public datasets and (ii) it encourages the network to learn local textural cues to robustly separate bonafide from fake fingerprints. This base CNN approach is illustrated in Figure 5.2 as the box enclosed by the green line. Importantly, this network is trained without any specific methods to improve cross-sensor and/or cross-material performance.

For most of our experiments, we utilize the MobileNet-v1 model [112] as our base CNN

architecture (the same as in [41]). To adapt the network for the two-class (live vs. PA) problem of fingerprint PAD, we replace the final 1000-unit softmax layer with a 2-unit softmax layer, and the network is trained with a batch size of 64 with an RMSProp optimizer. Additionally, to encourage robustness and avoid overfitting to minute variations of the input images, data augmentation tools of random distorted cropping, horizontal flipping, and random brightness were employed during training.

2.3.2 Adversarial Representational Learning (ARL)

ARL is an approach to domain generalization that aims to learn a generalized and robust feature representation for both source and target domains that performs well on predicting some target label while obfuscating the information related to a sensitive label. To accurately predict the target label in each domain, the goal of the "ARL" approach is to encourage an encoding network to learn a representation that is invariant between both domains. The ARL approach is shown in Figure 5.2 by the box enclosed by the red line. In the proposed approach, we incorporate two adversary networks with their own adversarial tasks: (i) sensor adversary and (ii) material adversary.

2.3.2.1 Sensor Adversarial Representational Learning (SARL)

SARL is used to learn a representation that is agnostic to which sensor generated the input fingerprint images (sensitive label), while accurately predicting live vs. PA (target label). In this setup, the encoder network is represented as a deterministic function, $\mathbf{z} = E(\mathbf{x}; \theta_{\mathbf{E}})$, the target prediction network estimates the conditional distribution $p(t|\mathbf{x})$ through $q_T(t|\mathbf{z}; \theta_{\mathbf{T}})$, and the sensor adversary network estimates the conditional distribution $p(s|\mathbf{x})$ through $q_{SA}(s|\mathbf{z}; \theta_{\mathbf{SA}})$, where $\mathbf{x}$ denotes the input fingerprint image, and $p(t|\mathbf{x})$ and $p(s|\mathbf{x})$ represent the probabilities of the target label (PA vs. Bonafide) t and sensitive label (sensor origin) s, respectively.

To learn this sensor-invariant representation, the sensor adversary network is trained to maximize the likelihood of predicting which sensor generated the input fingerprint image from the encoded representation. The parameters, $\theta_{\mathbf{SA}}$, of the sensor adversary network are updated to minimize the loss defined in equation 2.1.

Any other CNN-based approach other than [41] can be used instead.

$$\mathcal{L}_{SA} = \mathbb{E}_{\mathbf{x},s}[-\log q_{SA}(s|E(\mathbf{x};\theta_{\mathbf{E}});\theta_{\mathbf{SA}})] \tag{2.1}$$

2.3.2.2 Material Adversarial Representational Learning (MARL)

MARL is used to learn a representation that is agnostic to the fabrication material of a given PA sample. The loss to update the parameters of the material adversary is defined in Eq. 2.2, which is similar to the sensory adversary, except for replacing the sensitive task of predicting a given sample's sensor origin (s) to predicting its material composition (m).

$$\mathcal{L}_{MA} = \mathbb{E}_{\mathbf{x},m}[-\log q_{MA}(m|E(\mathbf{x};\theta_{\mathbf{E}});\theta_{\mathbf{MA}})] \tag{2.2}$$

The output of each adversary network is used to encourage the encoder to produce a representation that obfuscates the sensitive class labels by penalizing the parameters of the encoder, $\theta_{\mathbf{E}}$, to minimize the loss in equation 2.3, where α_S and α_M are hyper-parameters that allow for a trade-off between obfuscation of the sensitive labels and prediction of the target label. Meanwhile, to accurately predict bonafide vs. PA, the parameters of target prediction network, $\theta_{\mathbf{T}}$, are optimized to minimize the loss in equation 2.4.

$$\begin{aligned}\mathcal{L}_E = \mathbb{E}_{\mathbf{x},t}[-\log q_T(t|E(\mathbf{x};\theta_{\mathbf{E}});\theta_{\mathbf{T}})] + \alpha_s \mathbb{E}_{\mathbf{x}}[\sum_{i=1}^{m} q_{SA}(s_i|E(\mathbf{x};\theta_{\mathbf{E}});\theta_{\mathbf{SA}}) \log q_{SA}(s_i|E(\mathbf{x};\theta_{\mathbf{E}});\theta_{\mathbf{SA}})] \\ + \alpha_m \mathbb{E}_{\mathbf{x}}[\sum_{i=1}^{m} q_{MA}(m_i|E(\mathbf{x};\theta_{\mathbf{E}});\theta_{\mathbf{MA}}) \log q_{MA}(m_i|E(\mathbf{x};\theta_{\mathbf{E}});\theta_{\mathbf{MA}})]\end{aligned} \tag{2.3}$$

$$\mathcal{L}_T = \mathbb{E}_{\mathbf{x},t}[-\log q_T(t|E(\mathbf{x};\theta_{\mathbf{E}});\theta_{\mathbf{T}})] \tag{2.4}$$

2.3.3 Naïve

A simple approach to cross-sensor generalization is to assume access to a limited number of training examples (100 live and 100 PA fingerprint images) from the target sensor. This is a reasonable assumption in the case of cross-sensor generalization, where we have access to the sensing device on which the system will be deployed, and helps alleviate the necessity of collecting extensive amounts of data from the target domain. This is in contrast to generalization to unknown

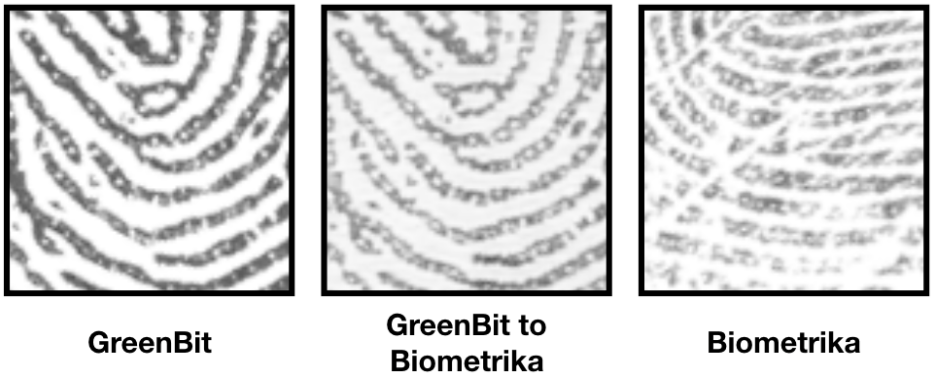


Figure 2.4 Example style transfer using the Universal Material Generator (UMG) to transfer the content of a source sensor minutiae patch (GreenBit) to the style of a target sensor minutiae patch (Biometrika).

PA materials, where we cannot assume any prior knowledge of the unknown target materials. We denote this method as the “naïve" approach to cross-sensor PAD as it does not require any changes to the system architecture.

2.3.4 Universal Material Generator (UMG)

The final aspect of our proposed approach is the incorporation of the style transfer network, which is used to augment the training data from the target sensor. The specific style transfer method we use is the Universal Material Generator (UMG) proposed in [44] that we have adapted for style transfer between sensor domains rather than between PA material types. Our version of the UMG accepts source and target domain minutiae patches as input and produces a large amount of synthetic training images in the target sensor domain. The style transfer is achieved through learning a mapping from the style of the source domain image patches to the style of the target domain image patches. Concretely, the network separates the content information, i.e, the fingerprint ridge structure, and the style, i.e, textural information, of a given fingerprint minutiae patch and produces a synthetic image that has the content of the source domain and the style of the target domain. An example of the style transfer between a content image from GreenBit to Biometrika is shown in Figure 2.4, and an overview of the “UMG" approach is shown as the box enclosed by the blue line in Figure 5.2.

2.3.5 UMG + SARL + MARL (Proposed Approach)

The proposed approach applies ARL (both SARL and MARL) with the UMG style transfer wrapper to further improve generalization performance. An illustration of the "UMG + SARL + MARL" approach is illustrated in Figure 5.2 as everything enclosed by the box formed by the solid, black line. Like the naïve approach, this method inherently assumes knowledge of a limited set of examples from the target domain sensor. Specifically, we assume 100 live and 100 PA images from the target sensor. From this small set of images from the target sensor, we produce a much larger set of synthetic images in the target domain using the UMG wrapper to transfer the style of the target domain to the content of the source domain training images.

The advantage of this approach is that we leverage the ability of the UMG wrapper to ensure a balanced dataset from all sensors (source and target), which we combine with ARL that forces the network to learn a sensor-invariant and material-invariant representation. In the following section, we demonstrate the performance gains over the previous approaches and show that the UMG coupled with ARL achieves the new state-of-the-art in cross-sensor and cross-material generalization for fingerprint PAD.

2.4 Evaluation Procedure

In this section we describe the experimental protocol of the various experiments carried out in this chapter, the datasets involved in each experiment, and the implementation details of the proposed approach.

2.4.1 Experimental Protocol

We adopt the leave-one-out protocol to evaluate cross-sensor PAD performance, where one sensor is set aside for testing, and the network is trained on data from all remaining sensors. The setup is then repeated for all combinations of source and target sensors. To separately study the impact of cross-sensor from cross-material performance, we divide our evaluation into two cases, one which includes all materials from the testing set in training (cross-sensor only), and the case where none of the testing materials were included in training (cross-sensor and cross-material).

Table 2.2 Summary of the 2015 and 2017 Liveness Detection (LivDet) datasets.

Dataset	**LivDet 2015**				**LivDet 2017**		
Fingerprint Reader	Green Bit	Biometrika	Digital Persona	CrossMatch	Green Bit	Orcanthus	Digital Persona
Model	DactyScan26	HiScan-PRO	U.are.U 5160	L Scan Guardian	DactyScan84C	Cerits2 Image	U.are.U 5160
Image Size	500 x 500	1000 x 1000	252 x 324	640 x 480	500 x 500	300 x $n^{\dagger}$	252 x 324
Resolution (dpi)	500	1000	500	500	569	500	500
#Live Images Train / Test	1000 / 1000	1000 / 1000	1000 / 1000	1510 / 1500	1000 / 1700	1000 / 1700	1000 / 1692
#PA Images Train / Test	1000 / 1500	1000 /1500	1000 / 1500	1473 / 1448	1200 / 2040	1180*/ 2018	1199 / 2028
PA Materials	Ecoflex, Gelatine, Latex, Wood Glue, Liquid Ecoflex, RTV, Body Double, PlayDoh, OOMOO				Wood Glue, Ecoflex, Body Double, Gelatine, Latex, Liquid Ecoflex		

† Fingerprint images captured by Orcanthus have a variable height (350 - 450 pixels) depending on the friction ridge content.

* A Set of 20 Latex PA fingerprints were present in the training data of Orcanthus; which were excluded in our experiments because only Wood Glue, Ecoflex, and Body Double are expected to be in the training dataset.

Furthermore, we analyze separately the cross-sensor scenarios where all train and test sensors employ the same sensing technology and when the training and test sensors employ different sensing technology. The following defines the three test scenarios:

1. *Cross-Sensor Only*: Training on all but one sensor and evaluating generalization performance on the left-out sensor. Test datasets contain only materials seen during training and each sensor employs the same sensing technology.
2. *Cross-Sensor and Cross-Material*: Training on all but one sensor and evaluating generalization performance on the left-out sensor. Test datasets contain only materials not seen during training and every sensor uses the same sensing technology.
3. *Cross-Sensing Technology*: Training on all but one sensor and evaluating generalization performance on the left-out sensor. Test datasets contain only materials seen during training and left-out sensor employs different sensing technology than the training sensors.

Table 2.3 Summary of the MSU-FPAD and GCT3 datasets.

Dataset	**MSU-FPAD**		**GCT3**[1]			
Fingerprint Reader	CrossMatch	Lumidigm	Optical A	Optical B	Optical C	Optical D
Model	Guardian 200	Venus 302	N/A	N/A	N/A	N/A
Image Size	750 x 800	400 x 272	1600 x 1600	180 x 256	500 x 500	500 x 468
Resolution (dpi)	500	500	500	385	500	1000
#Live Images Train / Test	2250 / 2250	2250 / 2250	10527 / 2632	6588 / 1647	6393 / 1599	8288 / 2072
#PA Images Train / Test	3000 / 3000	2250 / 2250	641 / 161	179 / 45	305 / 77	596 / 149
PA Materials	Ecoflex, PlayDoh, 2D Print (Matte Paper), 2D Print (Transparency)		Ecoflex, Gelatine, Silicone, Gummy Overlay with Conductive Silicone			

[1] Sponsor approval is required to release sensor name and model. Instead, descriptive identifiers are used based on the data types each sensor captures.

2.4.2 Datasets

The data used in the experiments are from the LivDet 2015, LivDet 2017, MSU-FPAD, and IARPA ODIN Government Control Test (GCT) datasets, which are summarized in Tables 2.2 and 2.3. The LivDet 2015 dataset consists of four sensors: Biometrika, CrossMatch, Digital Persona, and Green Bit. These sensors are all FTIR optical image capturing devices. We utilize this dataset to evaluate the generalization performance across different fingerprint readers with the same sensing technology. To evaluate performance on fingerprint readers with different sensing mechanisms, we experiment on fingerprint data from the Lumidigm sensor of the MSU-FPAD dataset. This sensor uses different sensing technology from the four seen in the LivDet 2015 as it is a multi-spectral, direct-view capture device. Furthermore, we incorporate a third dataset, LivDet 2017, which consists of three sensors: Digital Persona, Green Bit, and Orcanthus, where Orcanthus uses thermal-based imaging. Finally, a subset of four Optical-FTIR sensors from phase III of the IARPA ODIN sponsored Government Control Test data collection (GCT3) was also used.

2.4.3 Implementation Details

The architecture of the encoder in the proposed approach is MobileNet-v1 with the final 1000-unit softmax layer removed, which is used to encode a latent representation $\mathbf{z} \in \mathbb{R}^d$. In our implementation, $d = 1024$. The target predictor is a single fully connected layer of 2-dimensions (for predicting live vs. PA) with a softmax activation. The sensor (material) adversary network consists of a fully connected layer with a softmax activation of output dimension equal to the number of source sensors (PA materials) in the training dataset.

Training adversarial losses often requires extensive hyper-parameter tuning. For example, it was found advantageous during training to update the parameters, $\theta_{\mathbf{A}}$, of the adversary networks five times per every update of the encoder and target predictor. We also explored adjusting the number of hidden layers in the adversary networks, but no significant improvements over a single layer network were observed. A grid search was performed over the values of α_s and α_m for selecting the influence of the adversarial loss on updating the parameters, $\theta_{\mathbf{E}}$, of the encoder, and the optimal parameter values of $\alpha_s = 0.1$ and $\alpha_m = 0.05$ were selected for the sensor adversary and material adversary, respectively (see Eq. 2.3).

2.5 Experimental Results

Here we present the results of each experiment to evaluate the generalization performance of the proposed approach. This section is divided into several parts to facilitate an in-depth analysis of the generalization performance of the algorithm to each of the following cases: cross-sensor, cross-sensor and cross-material, and cross-sensing technology. A discussion on the effect of varying the number of assumed target domain images is included, as well as an analysis of the number of PA materials seen during training. We conclude this section with examining the deep feature space prior to and following the application of the proposed methodology for fingerprint PAD generalization. The feature space analysis is conducted utilizing a t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization [153].

There has not been much prior work aimed specifically at improving cross-sensor generalization of fingerprint PAD; nonetheless, there are a few cross-sensor performance results reported in the

Table 2.4 Cross-sensor generalization performance (TDR (%) @ FDR = 0.2 %) with leave-one-out protocol on LivDet 2015 dataset with materials common to training and testing, i.e., excluding cross-materials. Bio = Biometrika, CM = CrossMatch, DP = Digital Persona, and GB = GreenBit. The proposed method (UMG+SARL+MARL) provides the best cross-sensor generalization indicated by the highest mean TDR and small s.d. on the target sensor.

	Source CM, DP, GB	Target Bio	Source Bio, DP, GB	Target CM	Source Bio, CM, GB	Target DP	Source Bio, CM, DP	Target GB	**Source Mean± s.d.**	**Target Mean± s.d.**
Base CNN [112]	90.34	75.16	88.20	3.33	98.40	10.76	92.82	70.74	92.44 ± 4.40	40.00 ± 38.21
SARL	93.44	80.51	**91.03**	2.11	**98.73**	11.74	92.04	64.74	**93.81 ± 3.43**	39.78 ± 38.67
Naïve	87.74	84.80	88.23	97.37	96.96	59.13	88.08	90.68	90.25 ± 4.48	83.00 ± 16.72
UMG [44]	89.10	**94.33**	84.28	90.70	96.39	71.85	78.14	96.57	86.98 ± 7.71	88.36 ± 11.27
Naïve + SARL	90.18	91.86	87.87	98.95	94.21	52.07	89.15	83.92	90.35 ± 2.74	81.70 ± 20.69
UMG + SARL [92]	88.98	92.83	88.48	97.54	96.18	**87.61**	86.88	93.78	90.13 ± 4.13	92.94 ± 4.09
UMG + SARL + MARL	**96.22**	91.10	87.60	**100.0**	97.72	84.10	**93.15**	**96.90**	93.67 ± 4.47	**93.03 ± 7.00**

[1] We use FDR = 0.2 % because this is the stringent metric being used by the IARPA Odin program. Due to space limits, it is challenging to show the complete Receiver Operating Curve (ROC) or Detection Error Tradeoff (DET) curve.
[2] Liquid Ecoflex and RTV materials were excluded from the testing sets of Green Bit, Biometrika, and Digital Persona. Body Double, Playdoh, and OOMOO were excluded from the testing set of CrossMatch.

Table 2.5 Cross-sensor generalization performance (TDR (%) @ FDR = 0.2 %) with leave-one-out protocol on GCT3 dataset. A = optical sensor type A, B = optical sensor type B, C = optical sensor type C, and D = optical sensor type D.

	Source A, B, D	Target C	Source A, C, D	Target B	Source B, C, D	Target A	**Source Mean ± s.d.**	**Target Mean ± s.d.**
Base CNN [112]	89.95 ± 4.12	81.21 ± 7.03	90.95 ± 2.88	49.10 ± 14.15	86.48 ± 1.55	92.73 ± 5.60	89.13 ± 2.85	74.35 ± 8.93
UMG [44]	**96.34 ± 0.32**	91.42 ± 0.56	91.12 ± 2.65	68.02 ± 10.18	**91.81 ± 0.94**	**96.45 ± 0.46**	**93.09 ± 1.30**	85.30 ± 3.73
UMG + SARL [92]	95.80 ± 0.96	**92.23 ± 1.22**	92.64 ± 3.55	74.62 ± 8.31	90.75 ± 0.71	96.45 ± 0.76	93.06 ± 1.74	87.77 ± 3.43
UMG + SARL + MARL	95.80 ± 0.42	91.96 ± 0.95	**93.57 ± 3.07**	**78.68 ± 5.88**	89.21 ± 0.74	95.98 ± 0.08	92.86 ± 1.41	**88.87 ± 2.30**

literature. Chugh and Jain report the cross-sensor performance of Fingerprint Spoof Buster, which shares the same architecture of our base encoder model [41]. Therefore, in the following sections we can expect the performance against that of Fingerprint Spoof Buster to be similar to that of the Base CNN model. Furthermore, Chugh and Jain report cross-sensor results in their work toward

Table 2.6 Cross-sensor and cross-material generalization performance (TDR (%) @ FDR = 0.2 %) with leave-one-out protocol on LivDet 2015 dataset with materials exclusive to the testing datasets, i.e., cross-material only. Bio = Biometrika, CM = CrossMatch, DP = Digital Persona, and GB = GreenBit.

	Source CM, DP, GB	Target Bio	Source Bio, DP, GB	Target CM	Source Bio, CM, GB	Target DP	Source Bio, CM, DP	Target GB	**Source Mean± s.d.**	**Target Mean± s.d.**
Base CNN [112]	90.34	63.92	88.20	4.46	98.40	11.39	**92.82**	72.39	92.44 ± 4.40	38.04 ± 35.06
SARL	**92.78**	72.58	**91.03**	6.06	**98.73**	13.08	92.04	49.69	**93.65 ± 3.47**	35.35 ± 31.33
Naïve	87.74	77.11	88.23	96.80	96.96	42.62	88.08	85.69	90.25 ± 4.48	75.56 ± 23.39
UMG [44]	89.10	87.01	84.28	81.37	96.39	54.43	78.14	92.23	86.98 ± 7.71	78.76 ± 16.82
Naïve + SARL	90.18	86.19	87.87	97.45	94.21	35.65	82.51	65.44	88.69 ± 4.88	71.18 ± 27.15
UMG + SARL [92]	89.31	89.07	88.48	92.69	96.18	**78.69**	86.88	91.00	85.51 ± 7.00	87.86 ± 6.29
UMG + SARL + MARL	92.37	**90.20**	77.20	**99.54**	93.83	71.60	89.90	**92.60**	88.33 ± 7.59	**88.49 ± 11.93**

improving cross-material generalization with the introduction of their UMG network wrapper [44]. This approach is comparable to what we refer to as the UMG approach in following tables of this section.

2.5.1 Cross-Sensor Performance

To evaluate cross-sensor generalization we utilize the LivDet 2015 dataset and the GCT3 dataset which both consist of four different FTIR optical fingerprint imaging devices. The training and test sets of each GCT3 sensor contain exactly the same set of PA materials, whereas LivDet 2015 has additional PA materials in the testing sets. To separate out the cross-sensor generalization performance from the related task of cross-material generalization on the LivDet 2015 dataset, we first remove all the non-overlapping materials between the testing dataset of the target sensor and the training datasets of the three source sensors. For this experiment, Liquid Ecoflex and RTV materials were excluded from the testing sets when Green Bit, Biometrika, and Digital Persona were the target sensors, whereas, Body Double, Playdoh, and OOMOO were excluded from the

Table 2.7 Cross-sensor generalization performance (TDR (%) @ FDR = 0.2 %) on leave-out Biometrika (LivDet 2015) using Resnet-v1-50 as the base CNN model. Bio = Biometrika, CM = CrossMatch, DP = Digital Persona, and GB = GreenBit.

Test Dataset(s)	Base CNN [106]	SARL	Naïve	UMG [44]	Naïve + SARL	UMG + SARL [92]	UMG + SARL + MARL
Source: CM, DP, GB	65.29	72.72	73.55	72.76	73.05	75.94	**87.18**
Target: Bio	76.02	72.27	90.79	91.76	92.18	92.83	**93.50**

testing set with CrossMatch as the target sensor.

As shown in Tables 2.4 and 2.5, the proposed approach of UMG + SARL + MARL increases the average cross-sensor generalization in terms of True Detection Rate (TDR) at a False Detection Rate (FDR) of 0.2% from 88.36% to 93.03% over the UMG only method. The proposed approach also maintains higher performance (TDR = 93.67%) on the source domain sensors compared to the UMG only approach (TDR = 86.98%). Lastly, we note that the standard deviation (s.d.) across the four experiments of cross-sensor generalization on the LivDet 2015 dataset is significantly reduced for the UMG + SARL + MARL method (11.27% to 7.00%) in comparison to UMG only, indicating the robustness of the proposed approach.

For completeness, we include an evaluation of using an additional CNN architecture, Resnet-v1-50 [106], as the base encoder to demonstrate the generality of the proposed approach. In Table 2.7, we report the performance with ResNet-v1-50 as the Base CNN model on LivDet 2015 with leaving Biometrika out as the target sensor. We see that the performance improvement is consistent for both Base CNN models, supporting the generality of the approach to any existing CNN architecture trained for fingerprint PAD. In the remaining experiments, we continue to report results for only MobileNet-v1 as the Base CNN model.

We consider this metric to be more representative of actual use cases as opposed to EER and ACE. Space limitation does not allow us to show the full ROC curve.

Resnet-v1-50 was chosen since the authors of other SOTA fingerprint PAD algorithms were not willing to share their code and we found the details of their reported implementations insufficient for reproducing for a fair evaluation.

2.5.2 Cross-Sensor and Cross-Material Performance

In this section, we compare the performance of each solution on the cross-sensor and cross-material experiment by following the same procedure as the cross-sensor experiment, while including only materials exclusive to the test datasets of LivDet 2015. We saw that the adversarial learning improved the cross-sensor performance via enforcing a sensor-invariant representation, and we now evaluate whether we observe a similar benefit for the cross-material generalization (Table 2.6).

The results of Table 2.6 agree with the results of the cross-sensor only experiment shown previously; however, there is a small absolute performance decline due to the evaluation on only unknown PA materials. Specifically, the average TDR at a FDR of 0.2% of the proposed approach decreased from 93.03% for cross-sensor only to 88.49% for cross-sensor and cross-material generalization on the target sensor. However, we notice that the relative performance degradation of the UMG + SARL + MARL method is less than the relative drop in performance of the UMG only approach, which further demonstrates the generalization benefits of incorporating ARL for fingerprint PAD.

2.5.3 Cross-Sensing Technology Performance

In this section, we expand our analysis to include generalization across different fingerprint sensing mechanisms, where the sensing technology of the source fingerprint readers during training is different from the target test reader. For the first experiment we incorporate the data from the Lumidigm multispectral sensor of the MSU-FPAD database as the test sensor and the four FTIR optical sensors of LivDet 2015 as our training sensors. Here the testing datasets include only PA material types that were seen during training. The results show that UMG + SARL + MARL achieves the highest generalization TDR of 89.68% on the target domain sensor (Lumidigm), an improvement of 9.08% over the UMG only approach (Figure 2.8).

Figure 7.12 shows examples of bonafide and PA samples from Lumidigm that were correctly and incorrectly classified by the network trained with the proposed approach. Since the network averages predictions over minutiae patches extracted from each sample, the performance is dependent upon the reliability of the minutiae extraction. For example, the fingerprint ridge structure is ambiguous

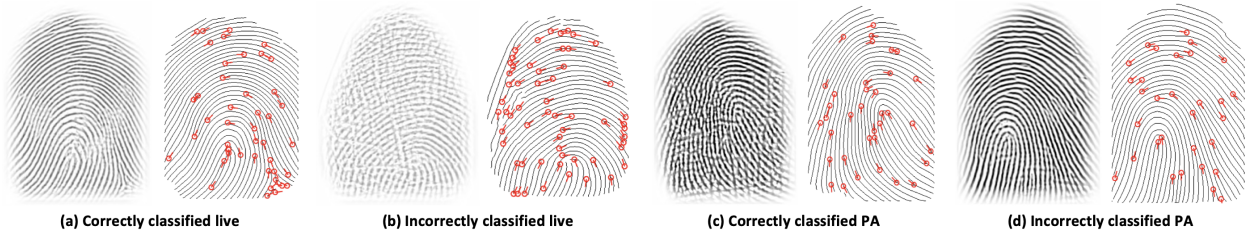


Figure 2.5 Example live and PA samples from the testing set of Lumidigm from MSU-FPAD that are correctly and incorrectly classified by a network trained via the proposed approach with Lumidigm left out of training.

Table 2.8 Cross-sensing technology generalization performance (TDR (%) @ FDR = 0.2 %) with four sensors of LivDet 2015 dataset included during training and Lumidigm from the MSU-FPAD dataset left out for testing. Bio = Biometrika, CM = CrossMatch, DP = Digital Persona, GB = GreenBit, and Lum = Lumidigm.

Test Dataset(s)	Base CNN [112]	SARL	Naïve	UMG [44]	Naïve + SARL	UMG + SARL [92]	UMG + SARL + MARL
Source: Bio, CM, DP, GB	**90.40**	87.41	63.54	88.24	87.22	88.45	86.70
Target: Lum	0.60	3.00	61.27	80.60	84.93	88.60	**89.68**

in many live samples that are incorrectly classified as PAs, whereas live samples with consistent, clear ridge structures tend to be correctly classified. To alleviate this dependence on the reliability of minutiae extraction, one could extract a minimum or fixed number of patches from each sample or even fuse the network predictions with a model trained on whole images, as was done by the authors of Spoof Buster [41].

2.5.4 Varying Number of Target Domain Images

To study the effect of varying the number of assumed target domain images available during training, we repeat the experiments in the leave-out Biometrika (LivDet 2015) scenario. Specifically, we run experiments on 50 and 250 live and PA training images from the target domain. As shown in Table 2.9, increasing the number of target domain images greatly benefits the naïve approach but only marginally affects the UMG + SARL method. Therefore, the benefit of UMG + SARL is most pronounced in cases with limited target domain training examples. In the trade-off between time spent for data collection and performance, the proposed method can significantly help reduce the burden of expensive data collection.

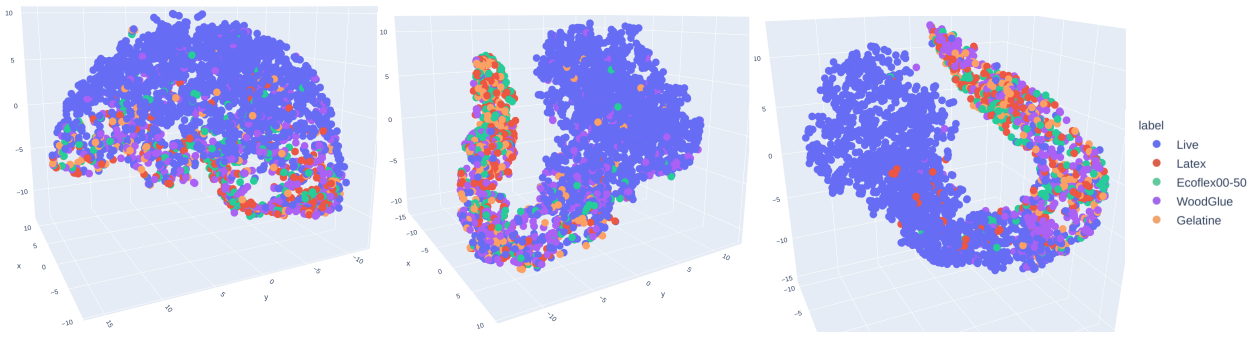
Table 2.9 Cross-sensor generalization performance (TDR (%) @ FDR = 0.2 %) on leave-out Biometrika (LivDet 2015) with varying number of target sensor training images.

Number of Target Domain Training Images	Test Dataset Domain	Naïve	UMG [44]	Naïve + SARL	UMG + SARL [92]
50	Source	91.21	**93.19**	85.64	90.76
	Target	90.15	90.47	91.43	**93.25**
250	Source	91.04	91.00	**95.50**	90.71
	Target	95.29	89.19	**95.40**	93.04

2.5.5 Varying PA Material Selection in Target Domain Images

In addition to varying the total number of live and PA images from the target sensor, we also experiment with varying the types of PA materials among those target sensor PA images that we include. Specifically, we first run the experiment with only including live images from the target sensor during training. Note, we still include all PA materials from the training sets for each of the source sensors. In other words, we are only varying the diversity of PA material types in the target domain that are seen during training. Then, we retrain while including two additional PA materials from the target domain (Ecoflex and Gelatine). Finally, we retrain again while including four PA materials in total (Ecoflex, Gelatine, Latex, and Wood Glue) from the target domain sensor.

We compare the source and target sensor generalization performance for each of these cases in Table 2.10. We observe that the target sensor performance is highest when including the most number of PA materials, where the difference is most pronounced in the case of the Digital Persona sensor as the target sensor. Plots of the TSNE feature space for each of the three networks on the test data from Digital Persona are shown in Figure 2.6, which show that the separation of bonafide and PA samples is greatest as we add more materials. On the other hand, the performance on the source domain sensors drops in two out of the three cases as we incorporate more PA data from the target domain into the training. This further highlights the trade-off we observed in the previous experiments between improved generalization performance at the expense of lower source domain performance.



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

Figure 2.6 3-dimensional t-SNE feature embeddings of the target sensor (Digital Persona) for networks trained via the proposed method on (a) live only, (b) live, ecoflex, and gelatine, and (c) live, ecoflex, gelatine, latex, and wood glue impressions from Digital Persona, in addition to the full training set of PA materials from GreenBit, Biometrika, and CrossMatch of LivDet 2015. The separation of bonafide vs. PA samples improves as the network is trained with more PA material types from the target sensor.

Table 2.10 Cross-sensor generalization performance of the UMG + SARL method (TDR (%) @ FDR = 0.2 %) on LivDet 2015 with varying number of target sensor PA materials during training. Bio = Biometrika, CM = CrossMatch, DP = Digital Persona, and GB = GreenBit.

	Source CM, DP, GB	Target Bio	Source CM, Bio, GB	Target DP	Source CM, DP, Bio	Target GB
Live Only[1]	**94.51**	90.80	91.88	77.50	**96.54**	88.3
Live, Ecoflex, Gelatine[2]	92.26	89.6	91.11	80.7	94.26	89.80
Live, Ecoflex, Gelatine, Latex, Wood Glue[3]	92.42	**94.60**	**96.70**	**90.10**	94.95	**92.50**

[1] All source sensor materials plus only Live images from the target sensor used during training.
[2] All source sensor materials plus Live, Ecoflex, and Gelatine images from the target sensor used during training.
[3] All source sensor materials plus Live, Ecoflex, Gelatine, Latex, and Wood Glue images from the target sensor used during training.

2.5.6 Effect of Different Training Sensor Technologies

Due to the large discrepancies between images, we analyze whether including additional data from vastly different sensing technology readers is beneficial or not. Intuitively, one would think that incorporating additional data should improve the predictions of the data-driven deep learning network; however, we show that this is not necessarily the case if the domain gap of the additional sensors is too large. To show this result, we train two networks on the LivDet 2017 dataset. The first network is trained with the GreenBit sensor designated as the target sensor, and the second network is trained with Digital Persona as the target sensor. Both GreenBit and Digital Persona

Table 2.11 Cross-sensor generalization performance (TDR (%) @ FDR = 0.2 %) with leave-one-out protocol for GreenBit and Digital Persona on LivDet2017 dataset with Orcanthus and without Orcanthus included in training. DP = Digital Persona, GB = GreenBit, and Orc = Orcanthus.

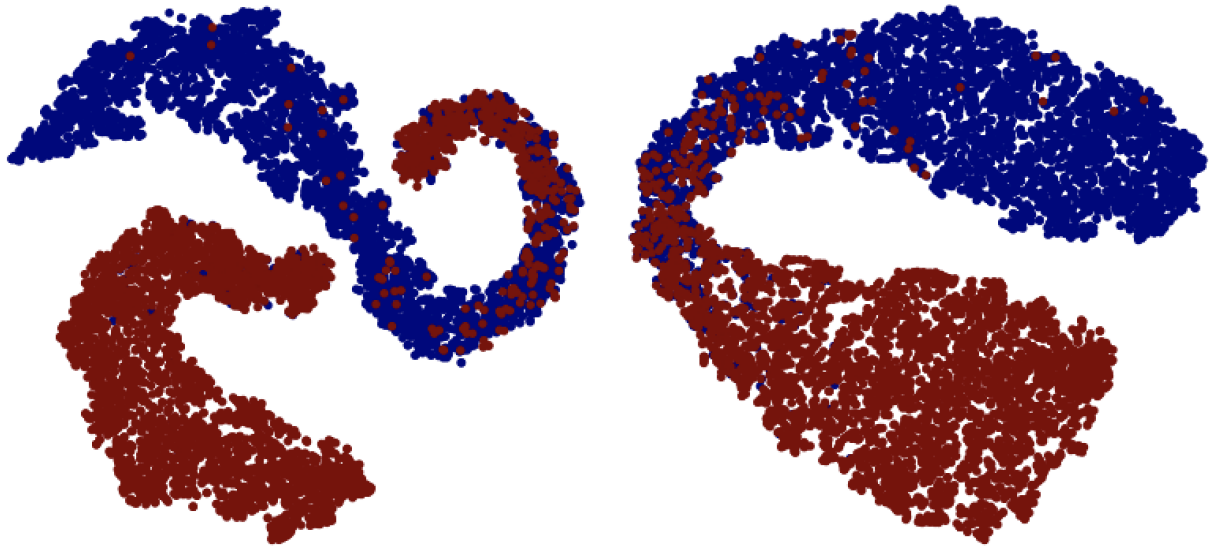
	Source DP (Orc)	Target GB	Source GB (Orc)	Target DP
UMG + SARL With Orc	5.50	24.19	21.31	25.85
UMG + SARL Without Orc	**10.24**	**59.19**	**75.66**	**32.55**

employ optical-FTIR based sensing, whereas the third sensor of LivDet 2017, Orcanthus, uses thermal-swipe based technology.

We first train each network on only images of the other optical-FTIR sensor in the dataset (GreenBit or Digital Persona) with the few samples of the target sensor as we did in all the previous experiments. Then, we compare the performance when incorporating training examples from the Orcanthus sensor. The results are shown in Table 2.11. We see that the performance for both the source and target domains is much better when Orcanthus is excluded from the training. This suggests that incorporating training data from sensors that are very different from your desired test sensors may actually degrade the performance on that test sensor.

2.5.7 Feature Space Analysis

To explore the benefits of incorporating ARL on top of the UMG only approach, we extract 2-dimensional t-SNE feature embeddings of the live and PA fingerprint minutiae patches from the final 1024-unit layer of the MobileNet-v1 encoder network, prior to the softmax non-linearity, from the UMG only network and the UMG + SARL network. For brevity, we just show the results of the leave-one-out protocol on the LivDet 2015 dataset with Biometrika, Green Bit, and Digital Persona as the source sensors and CrossMatch as the target sensor. In Figure 7.5, we plot these embeddings to analyze the effect of adversarially enforcing the learning of a sensor-invariant representation. Figure 7.5 (a) shows the separation between live and PA fingerprint minutiae patch embeddings of the UMG only network for minutiae patches from the target sensor, i.e., CrossMatch, whereas (b) shows the separation of the embeddings produced by the UMG + SARL approach. We can see that the proposed method provides noticeably better separation between the live and fingerprint PA



(a) (b)

Figure 2.7 2-dimensional t-SNE feature embeddings of the target sensor fingerprint minutiae patches for the (a) UMG only and (b) UMG + SARL models trained on the LivDet 2015 dataset with Biometrika, Green Bit, and Digital Persona as the source sensors and CrossMatch as the target sensor. The blue and red dots represent live and PA minutiae patches of fingerprint impressions captured on the target sensor (CrossMatch), respectively.

patches, resulting in the improved PAD performance.

2.6 Summary

Diverse and sophisticated presentation attacks pose a threat to the effectiveness of fingerprint recognition systems for reliable authentication and security. Previous PAD algorithms have demonstrated success in scenarios for which significant training data of bonafide and PA fingerprint images are available but are not robust enough to generalize well to novel PA materials unseen during training. Additionally, previous fingerprint PAD solutions are not generalizable across different fingerprint readers, meaning that a PAD algorithm trained on a specific fingerprint reader will not perform well when applied to different fingerprint sensing devices.

The approach towards fingerprint PAD presented in this chapter demonstrates an improvement over the state-of-the-art in terms of true detection rate (TDR) at a false detection rate (FDR) of 0.2% on cross-sensor and cross-material generalization. In particular, incorporating adversarial representation learning with the Universal Material Generator (UMG) improves the cross-sensor

generalization performance from a TDR of 88.36 ± 11.27% to 93.03 ± 7.00% on the LivDet 2015 dataset, while maintaining higher performance on the sensors seen during training. Furthermore, including cross-materials with the cross-sensor evaluation leads to an improvement of 78.76 ± 16.82% to 88.49 ± 11.93%. Lastly, experiments involving cross-sensing technology show average improvements of 80.60% to 89.68% with the proposed approach over state-of-the-art on the MSU-FPAD dataset. In the next chapter, we turn our attention to another case of cross-sensor fingerprint generalization, contact to contactless fingerprint matching, which has its own set of unique challenges stemming from the domain gap between contact and corresponding contactless fingerprint images.

2.7 Acknowledgment

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2017 – 17020200004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

CHAPTER 3

CONTACT TO CONTACTLESS FINGERPRINT MATCHING

Matching contactless fingerprints or finger photos to contact-based fingerprint impressions has received increased attention in the wake of COVID-19 due to the superior hygiene of the contactless acquisition and the widespread availability of low cost mobile phones capable of capturing photos of fingerprints with sufficient resolution for verification purposes. This chapter presents an end-to-end automated system, called C2CL, comprised of a mobile finger photo capture application, preprocessing, and matching algorithms to handle the challenges inhibiting previous cross-matching methods, namely i) low ridge-valley contrast of contactless fingerprints, ii) varying roll, pitch, yaw, and distance of the finger to the camera, iii) non-linear distortion of contact-based fingerprints, and vi) different image qualities of smartphone cameras. Our preprocessing algorithm segments, enhances, scales, and unwarps contactless fingerprints, while our matching algorithm extracts both minutiae and texture representations. A sequestered dataset of 9, 888 contactless 2D fingerprints and corresponding contact-based fingerprints from 206 subjects (2 thumbs and 2 index fingers for each subject) acquired using our mobile capture app is used to evaluate the cross-database performance of our proposed algorithm. Furthermore, additional experimental results on 3 publicly available datasets show substantial improvement in the state-of-the-art for contact to contactless fingerprint matching (TAR in the range of 96.67% to 98.30% at FAR=0.01%).

3.1 Introduction

Most prevailing fingerprint readers in use today necessitate physical contact of the user's finger with the imaging surface of the reader; however, this direct contact presents certain challenges in processing the acquired fingerprint images. Most notably, elastic human skin introduces a non-linear deformation upon contact with the imaging surface which has been shown to significantly degrade matching performance [13,35,201]. Furthermore, contact with the surface is likely to leave a latent impression on the imaging surface [191], which presents a security risk as an imposter

This chapter was previously published as S. A. Grosz, J. J. Engelsma, and A. K. Jain, "C2CL: Contact to Contactless Fingerprint Matching", IEEE Transactions on Information Forensics and Security, vol. 17, pp. 196-210, 2022.

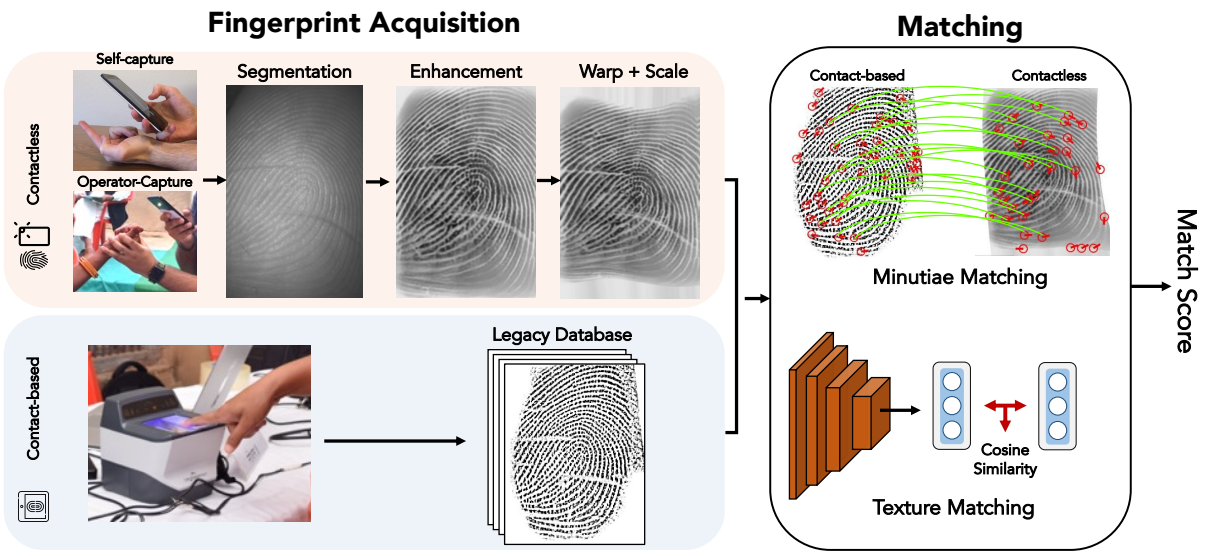


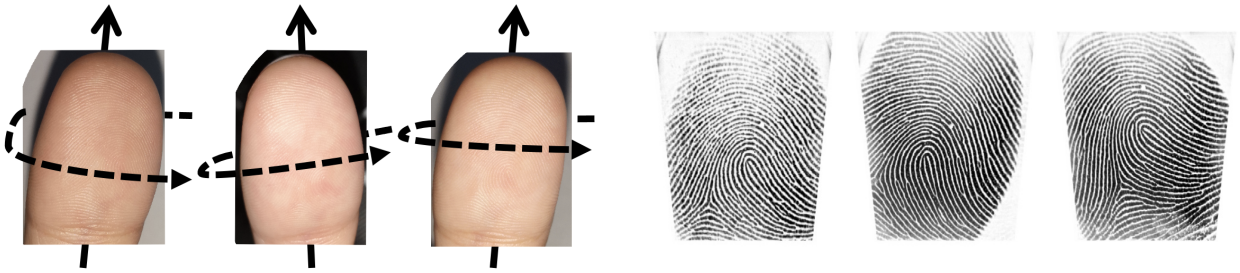
Figure 3.1 Overview of matching contactless fingerprint images with a legacy database of contact-based fingerprint impressions. While only a specific scenario is shown here where contact-based images are obtained from optical FTIR readers (slap or single finger capture) and contactless images are captured by a smartphone camera, our approach can be applied to any heterogeneous fingerprint matching problem.

could illegally gain access to the system though creation of a presentation (*i.e.*, spoof) attack.

In light of the ongoing Covid-19 pandemic, contactless fingerprint recognition has gained renewed interest as a hygienic alternative to contact-based fingerprint acquisition [183]. This is further supported by a recent survey that showed that the majority of users prefer touchless capture methods in terms of usability and hygeine considerations [194]. Prior studies have explored the use of customized 2D or 3D sensing for contactless fingerprint acquisition [110, 134, 136, 192, 220, 259], while others have explored the low-cost alternative of using readily available smartphone cameras to capture "finger photos" [157, 210, 224].

Despite the benefits of contactless fingerprint acquisition, imaging and subsequently matching a contactless fingerprint presents its own set of unique challenges. These include (i) low ridge-valley contrast, (ii) non-uniform illumination, (iii) varying roll, pitch, and yaw of the finger, (iv) varying background, (v) perspective distortions due to the varying distances of the finger from the camera, and (vi) lack of cross-compatibility with legacy databases of contact-based fingerprints

In general, contactless fingerprints refers to fingerprint images acquired by a contactless fingerprint sensor, whereas finger photo refers to fingerprint images acquired by a mobile phone. In this paper, we use the two terms interchangeably.



(a) (b)

Figure 3.2 Examples of contactless fingerprints (a) and their corresponding contact-based fingerprint images (b). Varying viewing angle, resolution, and illumination of contactless images and non-linear distortion of contact-based fingerprints contribute to the degradation of cross-matching performance. The contactless images shown are from the ZJU dataset.

(see Figure 3.2). For widespread adoption, contactless fingerprint recognition must overcome the aforementioned challenges and bridge the gap in accuracy compared to contact-contact fingerprint matching.

The most significant factor limiting the adoption of contactless fingerprint technology is cross-compatibility with legacy databases of contact-based fingerprints, which is particularly important for governmental agencies and large-scale national ID programs such as India's Aadhaar National ID program which has already enrolled over 1 billion users based upon contact-based fingerprints. Several studies have aimed at improving the compatibility of matching legacy slap images to contactless fingerprint images [49, 53, 145, 146, 158, 250]; however, none have achieved the same levels of accuracy as state-of-the-art (SOTA) contact-contact fingerprint matching (such as the results reported in FVC-ongoing [62] and NIST FpVTE [247]). Furthermore, all of these works focus on solving only a subset of the challenges in an effort to push the contact-contactless matching accuracy closer to the SOTA contact-contact matching systems. Indeed, to the best of our knowledge, this study presents the most comprehensive, end-to-end solution in the open academic literature for contact-contactless fingerprint matching that addresses the challenges inherent to each step in the contact to contactless matching process (mobile capture, segmentation, enhancement, scaling, non-linear warping, representation extraction, and matching).

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

Figure 3.3 Example contactless and contact-based fingerprint image pairs from databases which we have obtained from different research groups: (a) IIT Bombay [17], (b) ISPFDv2 [158], (c) MSU [53], (d) PolyU [145], (e) UWA [270], and (f) ZJU datasets. In general, contactless fingerprints suffer from low ridge-valley contrast, varying roll, pitch, and yaw, and perspective distortions, especially those captured by smartphone cameras (*e.g.*, (a), (b), (c) and (f)). We believe our study involves the largest collection of public domain databases of contactless and contact-based fingerprints.

We show that our end-to-end matcher, called C2CL, is able to significantly improve contact-contactless matching performance over the prevailing SOTA methods through experimental results on a number of different datasets, collected by various research groups using their own app and fingerprint readers. We also demonstrate that our matcher generalizes well to datasets which were not included during training. This cross-database evaluation solves a shortcoming of many existing studies which train and evaluate algorithms on different training and test splits of the same contact-contactless dataset. Furthermore, despite multiple evaluation datasets, we train only a single model for our evaluations, rather than fine-tuning individual models to fit a specific dataset.

Concretely, the contributions of this chapter are stated as the following:

1. An end-to-end system, called C2CL, for contact-contactless fingerprint matching. C2CL is comprised of preprocessing (segmentation, enhancement, scaling, and deformation correction), feature extraction (minutiae and texture representations), and matching modules. Our preprocessing also benefits the Verifinger 12.0 commercial fingerprint SDK.

Table 3.1 Summary of published cross-matching contact to contactless fingerprint recognition studies.

Study	Approach	Database	Accuracy†
Lin and Kumar, 2018 [145]	Robust TPS deformation correction model, minutiae and ridge matching	1,800 contactless and contact fingerprints from 300 fingers [145]. 2,000 contactless and 4,000 contact fingerprints from 1,000 fingers [270]	EER = 14.33% [145] EER = 19.81% [270]
Deb et al., 2018 [53]	COTS matcher	2,472 contactless and contact fingerprints from 1,236 fingers [53]	TAR = 92.4% – 98.6% @ FAR = 0.1% [53]
Lin and Kumar, 2019 [146]	Fusion of three Siamese CNNs	960 contactless and contact fingerprints from 160 fingers [145]. 1,000 contactless and 2,000 contact fingerprints from 500 fingers [270]	EER = 7.93% [145] EER = 7.11% [270]
Wild et al., 2019 [250]	Filtering based on NFIQ 2.0 quality measure, COTS matcher	1,728 contactless and 2,582 contact fingerprints from 108 fingers [250]	TAR = 95.5% – 98.6% @ FAR = 0.1% [250]
Dabouei et al., 2019 [49]	TPS spatial transformer network for deformation correction and binary ridge-map extraction network, COTS matcher	2,000 contactless and 4,000 contact fingerprints from 1,000 fingers [270]	EER = 7.71% [270]
Malhotra et al., 2020 [158]	Feature extraction with deep scattering network, random decision forest matcher	8,512 contactless and 1,216 contact fingerprints from 152 fingers [158]	EER = 2.11% – 5.23% [158]
Priesnitz et al., 2021 [194]	Neural network-based minutiae feature extraction, open-source minutiae matcher	896 contactless from two different capture setups and 464 contact fingerprints from 232 fingers [194]	EER = 15.71% and 32.02% [194]
Proposed Approach	TPS spatial transformer for 500 ppi scaling and deformation correction of contactless fingerprints. Fusion of minutiae and CNN texture representations.	8,512 contactless and 1,216 contact fingerprints from 152 fingers [158]. 2,000 contactless and 4,000 contact fingerprints from 1,000 fingers [270]. 960 contactless and contact fingerprints from 160 fingers [145]. 9,888 contactless and 9,888 contact fingerprints from 824 fingers (ZJU Dataset)	EER = 1.20% [158] EER = 0.72% [270] EER = 0.30% [145] EER = 0.62% (ZJU Dataset)

† Some studies only report EER while other studies only report TAR @ FAR = 0.1%.

2. A fully automated, preprocessing pipeline to map contactless fingerprints into the domain of contact-based fingerprints and a contact-contactless adaptation of DeepPrint [69] for representation extraction. Our preprocessing and representation extraction is generalizable across multiple datasets and contactless capture devices.
3. SOTA cross-matching verification and large-scale identification accuracy using C2CL on both publicly available contact-contactless matching datasets as well as on a completely

Table 3.2 Summary of contact to contactless fingerprint recognition datasets used in this study.

Dataset	# Subjects	# Unique Fingers	# Images (Contactless/Contact)	Contactless Capture Device	Contact Capture Device
UWA Benchmark 3D Fingerprint Database, 2014 [270]	150	1,500	3,000 / 6,000	3D Scanner (TBS S120E)	CROSSMATCH Verifier 300 LC2.0
ManTech Phase2, 2015 [75]	496	4,960	N/A / N/A*	AOS ANDI On-The-Go (OTG), MorphoTrak Finger-On-The-Fly (FOTF), IDair innerID on iPhone 4.	Cross Match Guardian R2, Cross Match SEEK Avenger, MorphoTrak MorphoIDent, MorphoTrust TouchPrint 5300, Northrop Grumman BioSled
PolyU Contactless 2D to Contact-based 2D Images Database, 2018 [145]	N/A	336	2,976 / 2,976	Low-cost camera and lens (specific device not given)	URU 4000
MSU Finger Photo and Slap Fingerprint Database, 2018 [53]	309	1,236	2,472 / 2,472	Xiaomi Redmi Note 4 smartphone	CrossMatch Guardian 200, SilkID (SLK20R)
IIT Bombay Touchless and Touch-Based Fingerprint Database, 2019 [17]	N/A	200	800 / 800	Lenovo Vibe k5 smartphone	eNBioScan-C1 (HFDU08)
ISPFDv2, 2020 [158]	76	304	17,024 / 2,432	OnePlus One (OPO) and Micromax Canvas Knight smartphones	Secugen Hamster IV
ZJU Finger Photo and Touch-based Fingerprint Database	206	824	9,888 / 9,888	HuaWei P20, Samsung s9+, and OnePlus 8 smartphones	URU 4500

* The number of contact and contactless images acquired per finger varies for each device; the exact number is not provided.

sequestered dataset collected at Zhejiang University, China. Our evaluation includes the most diverse set of contactless fingerprint acquisition devices, yet we employ just a single trained model for evaluation.

4. A smartphone contactless fingerprint capture app that was developed in-house for improved throughput and user-convenience. This app is made available to the public to promote further research in this area.
5. A new dataset of 9,888 2D contactless and corresponding contact-based fingerprint images from 206 subjects (2 thumbs and 2 index fingers per subject), which is made available to advance much needed research in this area.

The project repository for the smartphone contactless fingerprint capture app is available at https://github.com/ronny3050/FingerPhotos.
The dataset application is available at https://person.zju.edu.cn/en/eryunliu.

3.2 Prior Work

Prior studies on contact-contactless fingerprint matching primarily focus on only one of the sub-modules needed to obtain matching accuracy close to contact-contact based fingerprint matching systems (*e.g.*, segmentation, distortion correction, or feature extraction only). These studies are categorized and discussed below.

3.2.1 Segmentation

The first challenge in contact-contactless matching is segmenting the relevant fingerprint region from the captured contactless fingerprint images. Malhotra *et al*. [158] proposed a combination of a saliency map and a skin-color map to segment the distal phalange (*i.e.*, fingertip) of contactless fingerprint images in presence of varying background, illumination, and resolution. Despite impressive results, the algorithm requires extensive hyperparameter tuning and still fails to accurately segment fingerprints in severe illumination conditions or noisy backgrounds. To alleviate these issues, we incorporate segmentation via an autoencoder trained to robustly segment the distal phalange of input contactless images.

3.2.2 Enhancement

One of the main challenges with contactless fingerprint images is the low ridge-valley contrast (Figure 3.3). The literature has addressed this in a number of different ways, including adaptive histogram equalization, Gabor filtering, median filtering, and sharpening by subtraction of the Gaussian blurred image from the captured image ([49, 146, 158]). We also incorporate adaptive contrast enhancement in our work; however, one consideration that is lacking in existing approaches is the ridge inversion that occurs with Frustrated Total Internal Reflection (FTIR) optical imaging. In particular, the ridges and valleys of an FTIR fingerprint image will appear dark and light, respectively, while the opposite is true in contactless fingerprint images. Therefore, a binary inversion of the contactless fingerprint images is expected to improve the correspondence with their contact-based counterparts.

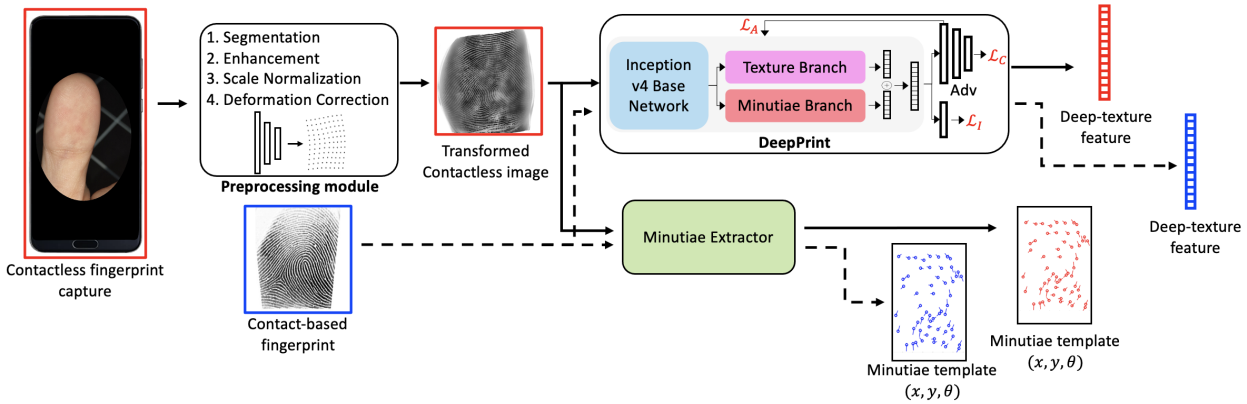


Figure 3.4 System architecture of C2CL. (a) A contactless fingerprint is captured and used as input to the preprocessing module, consisting of segmentation, enhancement, 500 ppi ridge frequency scaling, and deformation correction; (b) the transformed image output by the preprocessing module is fed to DeepPrint [69], which extracts a texture representation (shown in red). Without performing any additional preprocessing, the corresponding contact-based fingerprint is again fed to DeepPrint to extract a texture representation (shown in blue). Simultaneously, a minutiae representation is extracted using the Verifinger 12.0 SDK from both the contactless and contact-based fingerprint images.

3.2.3 Scaling

After segmenting and enhancing a contactless fingerprint, the varying distances between fingers captured and the camera must be accounted for. In particular, since contact-based fingerprints are almost always captured at 500 pixels per inch (ppi), the contactless fingerprints need to be scaled to be as close to 500 ppi as possible. Previous studies have applied a fixed manual scaling, set for a specific dataset, or have employed contact-based fingerprint ridge frequency normalization algorithms that rely on accurate ridge extraction - which is often unreliable for contactless fingerprints. In contrast, we incorporate a spatial transformer network [119] which has been trained to automatically normalize the resolution of the contactless fingerprints to match that of the 500 ppi contact images. This scaling is performed dynamically, *i.e.* every input contactless fingerprint image is independently scaled.

3.2.4 Distortion Correction

A final preprocessing step for contact-contactless matching is non-linear distortion correction. To address this problem, [145] used thin-plate-spline (TPS) deformation correction models (previ-

ously applied for contact-contact matching [13,47,201,202,213,215]) using the alignment between minutiae annotations of corresponding contactless and contact fingerprints. A limitation is that the transformation is limited to one of six possible parameterizations. In a different study, Dabouei et al. [47] train a spatial transformer to learn the distortion correction that is dynamically computed for each input image. In [47], a contact-based image is used as the reference for learning the distortion correction for a contactless image. However, we argue that this is not a reliable ground truth since the deformation varies among different contact-based fingerprint impressions. In our attempt to re-implement their algorithm, we found that this lack of a reliable and consistent ground truth makes training unstable, making it difficult to learn sound distortion parameters. In our work, rather than using the contact-based image as a reference, we use the match scores of our texture matcher as supervision for generating robust distortion correction. In other words, the distortion correction is optimized to maximize match scores between genuine contact-contactless fingerprint pairs.

3.2.5 Representation Extraction and Matching

After preprocessing a contactless fingerprint image to lie within the same domain as a contact-based fingerprint, a discriminative representation must be extracted for matching. In the prior literature there are two main approaches to feature representation: (i) minutiae representation ([47, 145]) and (ii) deep learning representation ([146, 158]). Minutiae-based approaches rely on clever preprocessing and other techniques to improve the compatability of contactless fingerprint images for traditional contact-based minutiae extraction algorithms. On the other hand, deep learning approaches place less emphasis on preprocessing to manipulate the contactless fingerprints to improve correspondence with contact-based fingerprints; rather, the responsibility is placed on the representation network to learn the correspondence despite the differences. For example, Lin and Kumar [144] and Dabouei et al. [47] both apply a deformation correction to the contactless image to improve the minutiae correspondence. In contrast, the deep learning approach taken in [146] applies very little preprocessing to the contactless image (just contrast enhancement and Gabor filtering) and leverages a Siamese CNN to extract features for matching. Similarly, Malhotra

et al. [158] utilize a deep scattering network to extract multi-scale and multi-directional feature representations.

In contrast to prior studies, our approach utilizes both a texture representation and a minutiae representation. Given the lower contrast and quality of contactless fingerprints (causing missing or spurious minutiae) and the non-linear distortion and scaling discrepancies between contact and contactless fingerprints (negatively impacting minutiae graph matching algorithms) a global texture representation is useful to improve the contact-contactless matching accuracy. We demonstrate this hypothesis empirically in the experimental results.

3.3 Methods

Our matcher, C2CL, aims to improve contact to contactless fingerprint recognition through a multi-stage preprocessing algorithm and matching algorithm comprised of both a minutiae representation and a texture feature representation. The preprocessing is employed to minimize the domain gap between the contactless fingerprints residing in a domain $\mathcal{D}_{cl}$ and contact-based fingerprints residing in another domain $\mathcal{D}_c$ and consists of segmentation, enhancement, ridge frequency scaling to 500 ppi, and deformation correction through a learned spatial transformation network. After preprocessing, we extract deep-textural and minutiae representations (unordered, variable length sets $T = \{(x_1, y_1, \theta_1), ..., (x_n, y_n, \theta_n)\}$) for matching. The final match scores are obtained via a score-level fusion between the texture and minutiae matching scores.

3.3.1 Preprocessing

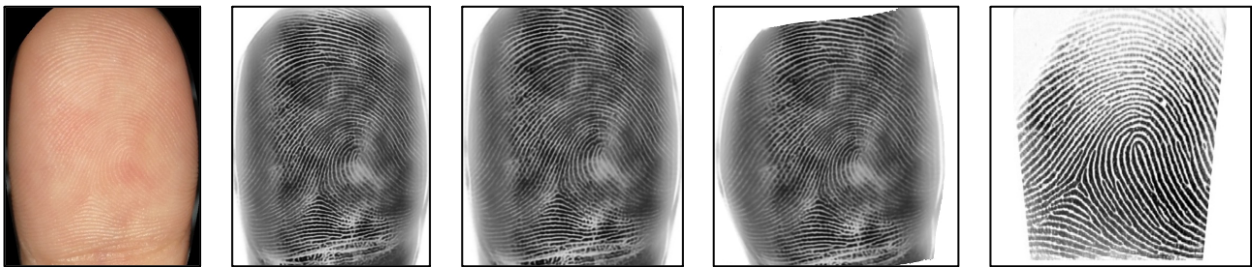
Here we discuss the details of each stage of our preprocessing algorithm as illustrated in Figure 3.6.

3.3.1.1 Segmentation

Many contactless fingerprint datasets are unsegmented; for example, the ISPFDv2 dataset [158] contains unsegmented, $(4,208 \times 3,120)$ images with varying illumination, resolution, and background conditions. Thus, the first step in our preprocessing pipeline is to segment the distal phalange of the fingerprint using a U-net segmentation network [200]. Our segmentation algorithm

(a) (b)

Figure 3.5 Example segmentation success (a) and failure (b) cases from images in the ISPFDv2 dataset using our segmentation algorithm. Sources of failure are presence of skin-like color tones in the background and varying skin complexion due to varying illumination.



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

Figure 3.6 Illustration of our preprocessing pipeline including (a) segmentation, (b) enhancement, (c) scaling, and (d) warping. For reference, a corresponding contact-based fingerprint is shown in (e).

is a network $S(\cdot)$ which takes as input the unsegmented contactless fingerprint I_{cl} of dimension $(m \times n)$ and outputs a segmentation mask $\hat{M} \in \{0, 1\}$ of dimension $(m \times n)$. The obtained segmentation mask, $\hat{M}$, is element-wise multiplied with I_{cl} to (i) crop out only the distal phalange of the contactless fingerprints and (ii) eliminate the remaining background to avoid detection of spurious minutiae in the later representation extraction stage. The segmented image I'_{cl} is then resized to 480×480 by maintaining the aspect ratio with appropriate padding for further processing.

For training $S(\cdot)$, we manually marked segmentation masks M of the distal phalange of 496 contactless fingerprints from the ISPFDv2 dataset. Initially, 200 images were randomly selected to have varying resolutions of either 5MP, 8MP, or 16MP and another 200 were selected with varying

We used the open source Labelme segmentation tool found on GitHub [243].

backgrounds and illumination. An additional 96 images were specifically selected for their greater perceived difficulty, particularly images with skin tone backgrounds. The optimization function for training $S(\cdot)$ is a pixel-wise binary cross-entropy loss between $\hat{M}$ and M (Eq. 3.1).

$$\mathcal{L}_{seg}(I_{cl}, I_c, M) = -\sum_{i,j}[M_{i,j}\log(\hat{M}_{i,j}|I_{cl}) + (1 - M_{i,j})\log(1 - \hat{M}_{i,j}|I_{cl})] \quad (3.1)$$

3.3.1.2 Enhancement

Following segmentation, we apply a series of image enhancements $E(\cdot)$ to increase the contrast of the ridge-valley structure of the contactless images, including (i) an adaptive histogram equalization to improve the ridge-valley contrast and (ii) pixel gray-level inversion to correct for the inversion of ridges between contact-based and contactless fingerprints. We also experimented with SOTA super resolution and de-blurring techniques, such as RDN [267], to further improve the contactless image quality but found only minimal matching accuracy improvements at the expense of significant additional computational cost.

3.3.1.3 Distortion Correction and Scaling

After segmenting and enhancing the contactless fingerprints, the non-linear distortions that separate the domains of contactless and contact-based fingerprints must be removed. In particular, this includes both a perspective distortion (caused by the varying distance of a finger from the camera) and a non-linear distortion (caused when the elastic human skin flattens against a platen).

To correct for these discrepancies, we train a spatial transformer network (STN) [119] $T(\cdot)$ that takes as input a segmented, enhanced contactless image $I^e_{cl} = E(I'_{cl})$ and aligns the ridge structure to better match the corresponding contact-based image domain $\mathcal{D}_c$. The goal of the STN is two-fold: (i) an affine transformation $T_s(\cdot)$ to normalize the ridge frequency of the contactless images to match the 500 ppi ridge spacing of the contact-based impressions and (ii) a TPS deformation warping $T_d(\cdot)$ of the contactless images to match the deformation present in contact-based images due to the elasticity of the human skin.

Both $T_s(\cdot)$ and $T_d(\cdot)$ are comprised of a shared localization network $l(\cdot, w)$ and individual

differentiable grid-samplers. Given an enhanced contactless fingerprint I^e_{cl}, $l(I^e_{cl}, w)$ outputs the scale (s), rotation (θ), and translation (t_x, t_y) of an affine transformation matrix A_s (Eq. 3.2) and a distortion field Θ which is characterized by a grid of $n \times n$ pixel displacements $\{(x_1, y_1)...(x_n, y_n)\}$. Subsequently, a scaled, warped image I^w_{cl} is obtained via Equation 3.3.

To learn the weights w of the localization network such that $T_s(\cdot)$ and $T_d(\cdot)$ correctly scale the contactless fingerprints to 500 ppi and unroll them into a contact-based fingerprint, we minimize the distance between DeepPrint representations extracted from genuine pairs of scaled, warped contactless fingerprints (I^w_{cl}) and contact-based fingerprints (I_c). In particular, let $f(\cdot)$ be a frozen DeepPrint network pretrained on contact-based fingerprints. Then, we can obtain a pair of 192D DeepPrint identity representations R_{cl} and R_c via $R_{cl} = f(I^w_{cl})$ and $R_c = f(I_c)$. Our loss can then be computed from Equation 3.4. By using the DeepPrint identity features extracted from contact-based fingerprint images to compute the loss, we are able to utilize the contact-based impressions as a ground truth of sorts. In particular, we are training our localization network to output better scalings and warpings such that the distortion and scale corrected contactless images have DeepPrint representations closer to their corresponding “ground truth" contact-based image.

We note that this approach has key differences to that which was proposed in [49] where the distortion corrected contactless image (scale was not learned in [49]) would be more directly compared to the ground-truth contact-based fingerprint via a cross-entropy loss between “binarized" versions of I^w_{cl} and I_c. We found that directly comparing the contactless and contact images via a cross entropy loss was quite difficult in practice since the ground truth contact image and the corresponding contactless image will have different rotations and translations separating them (even after scaling and distortion correction - resulting in a high loss value even if the scaling and distortion are correct). Furthermore, the contact-based image itself varies based upon the pressure applied during the acquisition, environmental conditions, sensor model, etc., meaning that directly using the contact-based image as ground truth is unreliable. In contrast, since DeepPrint has been trained to be invariant to pressure, environmental conditions, and sensor model, our ground truth (DeepPrint representations from contact-based images) will remain stable across different contact-

Table 3.3 Number of contactless and contact fingerprint images used in training each component of C2CL (No. contactless/No. contact).

Dataset	**Segmentation** $\mathbf{S}(\cdot)$	**Deformation Correction & Scaling** $\mathbf{T}(\cdot)$	**DeepPrint** $\mathbf{f}(\cdot)$
UWA Benchmark 3D Fingerprint Database [270]	0/0	0/0	1,000/2,000
ManTech Phase2, 2015 [75]	0/0	0/0	21,352/28,574
PolyU Contactless 2D to Contact-based 2D Images Database [145]	0/0	1,920/1,920	1,920/1,920
MSU Finger Photo and Slap Fingerprint Database [53]	0/0	2,472/2,472	2,472/2,472
IIT Bombay Touchless and Touch-based Fingerprint Database [17]	0/0	800/800	800/800
ISPFDv2 [158]	496/0	8,400/1,200	8,400/1,200
ZJU Finger Photo and Touch-based Fingerprint Database	0/0	0/0	0/0
Total	**496/0**	**13,592/6,392**	**35,944/36,966**

based impressions. In short, unlike [49], we learn both distortion correction **and** scaling correction simultaneously, and we use the DeepPrint identity loss to stabilize training of $T(\cdot)$ and to enable predictions of warpings and scalings which better improve matching accuracy.

$$A_s = \begin{bmatrix} s\cos(\theta) & -s\sin(\theta) & t_x \\ s\sin(\theta) & s\cos(\theta) & t_y \end{bmatrix} \tag{3.2}$$

$$I_{cl}^w = T(I_{cl}^e; A_s, \Theta) = T_d(T_s(I_{cl}^e, A_s), \Theta) \tag{3.3}$$

$$\mathcal{L}_{STN} = \|R_{cl} - R_c\|_2^2 \tag{3.4}$$

3.3.2 Representation Extraction

After performing all of the aforementioned preprocessing steps, we enter the second major stage of our contact-contactless matcher, namely the representation extraction stage. Our representation extraction algorithm extracts both a textural representation (using a CNN) and a minutiae-set. Scores are computed using both of these representations and then fused together using a sum score fusion.

3.3.2.1 Texture Representation

To extract our textural representation, we fine-tune the DeepPrint network proposed by Engelsma et al. in [69] on a training partition of the publicly available datasets which we aggregated (Table 3.3). Unlike the deep networks used in [146] and [158] for extraction of textural representations, DeepPrint is a deep-network which has been specifically designed for fingerprint representation extraction via a built-in alignment module and minutiae domain knowledge. Therefore, in this work, we seek to adopt DeepPrint for contact-contactless fingerprint matching. As is common practice in the machine learning and computer vision communities, we are utilizing a pretrained DeepPrint network to warm start our model, which has been shown to improve over random initialization for many applications (for example, in fingerprint spoof detection [177]).

Formally, DeepPrint is a network $f(\cdot)$ with parameters w that takes as input a fingerprint image I and outputs a fixed-length fingerprint representation R (which encodes the textural related features). During training, DeepPrint is guided to encode features related to fingerprint minutiae via a multi-task learning objective including (i) a cross-entropy loss on both the minutiae branch identity classification probability $\hat{y}_1$ and texture branch identity classification probability $\hat{y}_2$ (Eq. 3.5), (ii) minimize the intra-class variance of class y via a center loss between the predicted minutiae feature vector R_1 and its mean feature vector R_1^y and the predicted texture feature vector R_2 and its mean feature vector R_2^y (where R_1 concatenated with R_2 form the full representation R), and (iii) a mean squared error loss on the predicted minutiae maps $\hat{H}$ output by DeepPrint's minutiae branch and ground truth minutiae maps H (Eq. 3.7). These losses are combined to form the DeepPrint identity loss, $\mathcal{L}_{ID}$ (Eq. 3.8), where $\lambda_1 = 1$, $\lambda_2 = 0.00125$, $\lambda_3 = 0.095$ are set empirically.

$$\mathcal{L}_1(I, y) = -\log(\hat{y}_1^{j=y}|I, w) - \log(\hat{y}_2^{j=y}|I, w)] \tag{3.5}$$

$$\mathcal{L}_2(I, y) = \|R_1 - \bar{R}_1^y\|_2^2 + \|R_2 - \bar{R}_2^y\|_2^2 \tag{3.6}$$

$$\mathcal{L}_3(I, H) = \sum_{j,k,l} (\hat{H}_{j,k,l} - H_{j,k,l})^2 \tag{3.7}$$

$$\mathcal{L}_{ID}(I, y, H) = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} [\lambda_1 \mathcal{L}_1(I^i, y^i) + \lambda_2 \mathcal{L}_2(I^i, y) + \lambda_3 \mathcal{L}_3(I^i, H^i)] \tag{3.8}$$

Due to the differences in resolution, illumination, and backgrounds observed between different datasets of contactless fingerprint images, generalization to images captured on unseen cameras becomes critical. The problem of cross-sensor generalization in fingerprint biometrics (*e.g.*, optical reader to capacitive reader), of which contact to contactless matching is an extreme example, has been noted in the literature [4, 5, 152, 203], with many previous works aimed at improving the interoperability [123, 161, 204]. Motivated by the recent work employing adversarial learning to cross-sensor generalization of fingerprint spoof detection [92], we incorporate an adversarial loss to encourage robustness of DeepPrint to differences between acquisition devices. The adversarial loss $\mathcal{L}_A$ is defined as the cross-entropy on the output of an adversary network $q(\cdot, \theta_A)$ across C classes of sensors, where the adversarial ground truth y' is assigned equal probabilities across these C classes (Eq. 3.9). The adversarial loss $\mathcal{L}_A$ and identity loss $\mathcal{L}_{ID}$ form the overall loss function $\mathcal{L}_D$ used to train DeepPrint (Eq. 3.10), where $\lambda_4 = 0.1$ is empirically selected. The adversary network, $q(\cdot, \theta_A)$, is a two layer fully connected network, with weights θ_A, that predicts the probability of the class of input device used to capture each image, *i.e.*, minimizes the cross-entropy of the predicted device and the ground truth device label y (Eq. 3.11). Intuitively, if DeepPrint learns to fool the adversary, it has learned to encode identifying features which are independent of the acquisition device or camera.

$$\mathcal{L}_A(I, y') = -\sum_{c=1}^{C} y'_c \log q_A(y_c | f(I; w); \theta_A) \tag{3.9}$$

$$\mathcal{L}_D(I, y, H, y') = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} [\mathcal{L}_{ID}(I, y, H) + \lambda_4 \mathcal{L}_A(I^i, y'^i)] \tag{3.10}$$

$$\mathcal{L}_C(I, y_c) = -y_c \log q_A(y_c | f(I; w); \theta_A)] \tag{3.11}$$

In addition to the adversarial loss, we also increased the DeepPrint representation dimensionality from the original 192D to 512D and added perspective distortion and scaling augmentations during training. In an ablation study (Table 3.8), we show how each of our DeepPrint modifications (fine-tuning, adversarial loss, perspective and scaling augmentations, and dimensionality change) improves the contact-contactless fingerprint matching performance.

3.3.2.2 Minutiae Representation

Finally, after extracting a textural representation with our modified DeepPrint network, we extract a minutiae-based representation from our preprocessed contactless fingerprints with the Verifinger 12.0 SDK.

3.3.3 Matching

Following feature extraction, from which we obtain texture representations (R_t^c, R_t^{cl}) and Verifinger minutiae representations (R_m^c, R_m^{cl}) for a given pair of contact and contactless fingerprint images (I_c, I_{cl}), we compute a final match score as a weighted fusion of the individual scores computed between (R_t^c, R_t^{cl}) and (R_m^c, R_m^{cl}). Concretely, let s_t denote the similarity score between (R_t^c, R_t^{cl}) and s_m denote the similarity score between (R_m^c, R_m^{cl}), then the final similarity score is computed from a sum score fusion shown in Equation 3.12. For our implementation, $w_t = w_m = 0.5$ was selected empirically.

$$s = w_t s_t + w_m s_m \tag{3.12}$$

3.4 Experiments

In this section, we give details on various experimental evaluations to determine the effectiveness of C2CL for contact to contactless fingerprint matching. We employed various publicly available datasets for the evaluation of our algorithms, as well as a new database of contactless and corresponding contact-based fingerprints which was collected using our mobile-app in coordination with Zhejiang University (ZJU).

3.4.1 Datasets

Table 7.1 gives a detailed description of the publicly available datasets for contact to contactless matching used in this study, and Figure 3.3 shows some example images from these datasets. For comparison with previous studies, we use the same train/test split of the PolyU dataset that was used in [146], which consists of 160 fingers for training with 12 impressions each and the remaining 160 fingers for testing with 6 impressions each. Similarily, we split the UWA Benchmark 3D dataset into 500 training fingers and $1,000$ unique test fingers. Furthermore, following the protocol of Malhotra et al. [158], we split the ISPFDv2 dataset evenly into 50% train and 50% test subjects. Finally, we captured and sequestered a new dataset of contactless fingerprints and contact-based fingerprint images in coordination with ZJU for a cross-database evaluation (*e.g.*, not seen during training) to demonstrate generalizability of our algorithm. The cross-database evaluation is much more stringent than existing approaches which only train/test on different partitions of the same dataset. Indeed, the cross-database evaluation is a much better measure of how C2CL would perform in the real world.

The ZJU Finger Photo and Touch-based Fingerprint Database contains a total of 206 subjects, with 12 contactless images and 12 contact-based impressions per finger. The thumb and index fingers of both hands were collected for each subject, giving a total of $9,888$ contactless and contact-based images each. The contactless images were captured using three commodity smartphones: HuaWei P20, Samsung s9+, and OnePlus 8, whereas the contact-based fingerprint impressions were captured on a URU 4500 optical-based scanner at 512 ppi. An Android fingerphoto capture app was developed to improve the ease and efficiency of the data collection. To initiate the capturing process, a user or operator enters the transaction ID for the user and uses an on screen viewing window to help guide and capture the fingerprint image. Furthermore, a counter displayed on the screen keeps track of subsequent captures to streamline the data collection process.

3.4.2 Implementation Details

All the deep learning components (segmentation network, deformation correction and scaling network, and DeepPrint) are implemented using the Tensorflow deep learning framework. Each

network is trained independently and information regarding how many of the contactless and contact fingerprint images from each of the datasets used in training each component of our algorithm is given in Table 3.3.

3.4.2.1 Segmentation Network

A total of 496 contactless fingerprint images from the ISPFDv2 were manually labeled with segmentation masks outlining the distal phalange were used for training. Input images were down sampled to 256×256 during training to reduce the time to convergence, which occurred around $100,000$ iterations using stochastic gradient descent (SGD) with a learning rate of $1e^{-3}$ and a batch size of 8 on a single NVIDIA GeForce RTX 2080 Ti GPU. During inference, the contactless fingerprint images are resized to 256×256 and resulting segmentation masks are upsampled back to the original resolution. Due to limited number of manually marked images, we employed random rotations, translations, and brightness augmentations to avoid over-fitting. Additionally, we incorporated random resizing of input training images within the range $[128 \times 128, 384 \times 384]$ to encourage robustness to varying resolution between capture devices.

3.4.2.2 Deformation Correction and Scaling Network

The pretrained DeepPrint model in [69] was used to provide supervision of our spatial transformation network in line with Eq 3.4. The motivation for using a network pretrained on contact-based fingerprints, rather than our new finetuned model on contactless fingerprints, is that the goal of our transformation network is to transform the contactless fingerprint images to better resemble their contact-based counterparts. Thus, a supervisory network trained on solely contact-based fingerprint images is more suitable for this purpose. The architectural details of our STN localization network are given in Table 3.4. For our implementation, we set the number of sampling points for the distortion grid to n = 4×4. Data augmentations of random rotations, translations, brightness adjustments, and perspective distortions were employed to avoid over-fitting. This network was trained for $25,000$ iterations using an Adam optimizer with a learning rate of $1e^{-6}$ and a batch size of 16 on a single NVIDIA GeForce RTX 2080 Ti GPU.

Table 3.4 Deformation correction and scaling spatial transformation network architecture, $T(\cdot)$.

Layer	#Filters, Filter Size, Stride	Output Dim.
0. Input	0, 0, 0	$480 \times 480 \times 1$
1. Convolution	$32, 3 \times 3, 2$	$240 \times 240 \times 32$
2. Convolution	$64, 3 \times 3, 2$	$120 \times 120 \times 64$
3. Convolution	$128, 3 \times 3, 2$	$60 \times 60 \times 128$
4. Convolution	$256, 3 \times 3, 2$	$30 \times 30 \times 256$
5. Max Pool	$256, 6 \times 4, 2$	$13 \times 14 \times 256$
6. Dense	–	1024
7. Dense	–	$2 \times n_0 + 4$

The final dense layer contains output neurons for a $2 \times n_0$ grid of $n_0 = n \times n$ pixel displacements and 4 neurons for the affine transformation matrix (s, θ, t_x and t_y.). In our implementation, $n = 4$.

3.4.2.3 DeepPrint

The DeepPrint network was trained on two NVIDIA GeForce RTX 2080 Ti GPUs with an RMSProp optimizer, learning rate of 0.01, and a batch size of 16. The added adversary network, which was trained in step with DeepPrint, also utilized an RMSProp optimizer with a learning rate of 0.01. A small validation set was partitioned from the DeepPrint fine-tuning data outlined in Table 3.3 to stop the training (which occurred at 73,000 steps). Lastly, random rotation, translation, brightness, cropping, and perspective distortion augmentations were utilized during training.

3.4.3 Evaluation Protocol

To evaluate the cross-matching performance of our algorithms, we conduct both verification (1:1) and identification (1:N) experiments. For verification, we report the Receiver Operating Characteristic (ROC) curves at specific operating points and equal error rates (EER). Note that we report the True Acceptance Rate (TAR) at a False Acceptance Rate (FAR) of 0.01%, which is a stricter threshold than is currently reported in the literature, and which is also a threshold expected for field deployment. For the search experiments, the rank-one search accuracy is given against an augmented large scale gallery of 1.1 million contact fingerprints taken from an operational forensics database [262]. This is a much larger gallery than has previously been evaluated against in the literature and is again more indicative of what C2CL would face in the real world. Finally, we present ablation results on each significant component of our proposed system.

Table 3.5 Verification performance comparison.

Dataset	Verifinger 12.0		DeepPrint		DeepPrint + Verifinger 12.0		Previous SOTA
	EER (%)	TAR (%) @ FAR=0.01%	EER (%)	TAR (%) @ FAR=0.01%	EER (%)	TAR (%) @ FAR=0.01%	EER (%)
PolyU	0.46	97.20	2.37	72.07	0.30	97.74	7.93 [146]
UWA	6.81	92.56	5.29	83.40	0.72	98.30	7.11 [146]
ISPFDv2	1.46	96.02	2.33	84.33	1.20	96.67	3.40[1][158]
ZJU[2]	0.79	96.86	2.08	86.42	0.62	97.56	N/A

[1] [158] reports results on the ISPFDv2 dataset per individual capture condition; 3.40 is the average EER across these data splits.
[2] Cross-database evaluation, *i.e.*, not seen during training.

3.4.4 Verification Experiments

The verification experiments are conducted in a manner consistent with previous approaches to facilitate a fair comparison. In particular, (i) the PolyU testing dataset yields $5,760$ ($160 \times 6 \times 6$) genuine scores and $915,840$ ($160 \times 159 \times 6 \times 6$) imposter scores, (ii) the UWA Benchmark 3D dataset yields $8,000$ ($1,000 \times 4 \times 2$) genuine and $7,992,000$ ($1,000 \times 999 \times 4 \times 2$) imposter scores, (iii) the ISPFDv2 dataset (which is split into 7 different capture variations) yields $68,096$ ($(152 \times 8 \times 8) \times 7$) genuine and $10,282,496$ ($(152 \times 151 \times 8 \times 8) \times 7$) imposter scores, and (iv) the ZJU dataset yields $118,656$ ($824 \times 12 \times 12$) genuine and $97,653,888$ ($824 \times 823 \times 12 \times 12$) imposter scores. Due to the very high number of possible imposter scores for ZJU, we limit the number of imposter scores computed to only include the first impression of each imposter fingerprint. This process results in $678,152$ imposter scores out of the possible $97,653,888$ scores. It is assumed for all experiments that the contactless fingerprints and contact-based impressions are the probe and enrollment images, respectively.

Table 3.5 provides the Equal Error Rate (EER) and TAR @ FAR=0.01% of C2CL on the different datasets. For comparison with previous methods, rather than implement the relevant SOTA approaches that have been proposed and risk under representing those methods, we directly compare our approach to the results reported in each of the respective papers. In terms of EER, our method outperforms all the previous approaches in the verification setting. Not only does our

The 7 scenarios consist of different background, illumination, and resolution variations (*e.g.*, white background & indoor lighting, white background & outdoor lighting, natural background & indoor lighting, natural background & outdoor lighting, 5MP resolution, 8MP resolution, and 16MP resolution. For our evaluation, we combine each of these into a single dataset.

Table 3.6 DeepPrint verification performance (finetuned on PolyU dataset only).

Test Dataset	**EER (%)**	**TAR (%) @ FAR=0.01%**
PolyU	1.90	74.62
UWA	8.35	35.42
ISPFDv2	3.87	57.10
ZJU	2.99	68.99

individual performance of the minutiae and textural representations alone exceed that of the previous SOTA methods (in particular, even if we remove Verifinger, we still beat SOTA in all cases), the fusion performance attains matching accuracy (EER = 0.30% − 1.20%), which is much closer to contact-contact fingerprint matching [62]. Even in the most challenging cross-database evaluation (ZJU), C2CL attains competitive performance with contact-contact matching, demonstrating the generalizability of C2CL to unseen datasets. Note that we report the TAR @ FAR=0.01% only for C2CL since most of the prior approaches only report EER and none report TAR @ FAR =0.01%.

Different from previous approaches, which train individual models on a train/test split for each evaluation dataset, we have trained just a single model for our evaluation across four different datasets. This protocol is actually more challenging than finetuning for each individual evaluation dataset. This is because despite having a smaller number of training samples, higher verification performance can more easily be achieved by individually trained models. To support this claim, we have finetuned an additional model on just the PolyU dataset using the same train/test split specified in [146] and recorded the verification performance in Table 3.6. We observe that our accuracy improves on PolyU from 2.37% EER for the model trained on our full combination of training datasets to 1.90% EER for the model trained on just PolyU; however, because of the lower performance on the other three datasets, we can see that this model is indeed over-fit to PolyU.

3.4.4.1 Ablation study

We present an ablation study (Table 5.9) to fully understand the contribution of the main components of our algorithm; namely, segmentation, enhancement, 500 ppi frequency scaling, and TPS deformation correction. From the ablation, we notice there is a substantial improvement in

both EER and TAR @ FAR=0.01% just from incorporating proper enhancement of the contactless images. In most cases, there is almost a 50% reduction in EER from including both contrast enhancement and binary pixel inversion. For brevity, not shown in the table is the individual contribution of inverting the ridges of the contactless images aside from contrast enhancement. For reference, the EER of DeepPrint on ZJU warped images with only contrast enhancement is 2.49%. This is in comparison to the EER of 2.08% on the warped images with both contrast enhancement and pixel inversion.

Furthermore, we observe that for the smartphone captured contactless fingerprints in the ISPFDv2 and ZJU datasets, there is a dramatic performance jump when incorporating our 500 ppi scaling network. Additionally, there is another noticeable improvement when incorporating the deformation correction branch of our STN, most notably for the ISPFDv2 dataset. Since the ZJU dataset contains equal numbers of thumb and index fingers, where the majority of our training datasets contain mostly non-thumb fingers, we observed that the deformation correction is less beneficial on average for the ZJU dataset compared to ISPFDv2. In fact, from Table 3.9, we see that the EER of just index fingers of ZJU is noticeably lower than the EER on thumbs.

To investigate whether the lower performance on thumbs is a limitation of the available training data or whether thumbs require a different distortion correction from non-thumbs, we retrained separate warping models on thumb data only and non-thumb data only. The test results for the ZJU dataset are in Table 3.10. A couple of observations: i.) The performance (TAR @ FAR=0.01%) is highest for the model trained on both thumbs and non-thumbs, ii.) the model trained on non-thumbs performs slightly worse when applied to the test set of thumbs in the ZJU dataset, which indicates that the warping required for thumbs may be slightly different, and iii.) the performance of the thumb only model decreases on both thumbs and non-thumbs due to the limited number of thumb training examples.

3.4.4.2 Multi-finger fusion verification

The final set of verification experiments is to investigate the effects of finger position and multiple finger fusion in the verification accuracy for the ZJU dataset. Table 3.9 shows the

Table 3.7 Ablation study of C2CL using only Verifinger 12.0 for matching*. S = segmentation, E = enhancement, T_s = scaling, T_d = deformation correction.

Dataset	S	E	T_s	T_d	EER (%)	TAR @ FAR = 0.01%
PolyU	✓				0.86	93.19
	✓	✓			0.45	96.96
	✓	✓	✓		0.48	96.44
	✓	✓	✓	✓	0.46	97.20
UWA‡	✓				6.62	91.05
	✓	✓			6.81	92.56
ISPFDv2	✓				13.76	23.93
	✓	✓			7.83	38.53
	✓	✓	✓		2.02	93.3
	✓	✓	✓	✓	1.46	96.02
ZJU†	✓				3.35	82.8
	✓	✓			1.88	89.9
	✓	✓	✓		0.90	96.97
	✓	✓	✓	✓	0.79	96.86

* Ablation results for DeepPrint are not shown since only a single model was trained on the final E+S+T_s+T_d images.
‡ We do not apply our STN here since these images are captured with a 3D scanner and are already unrolled and at a resolution of 500 ppi.
† Cross-database evaluation, *i.e.*, not seen during training.

individual performance per finger position and the fusion of multiple fingers; namely, thumb only, index only, fusion of right thumb and right index, fusion of left thumb and left index, and four finger fusion. The motivation for considering fusion of the thumb and index on each hand is that from a usability standpoint, a user may be able to use their dominant hand when capturing their own fingerprints. Notably, when fusing multiple fingers (*e.g.*, right index and left index), we obtain nearly perfect accuracy.

Table 3.8 DeepPrint ablation study

Method	**ZJU EER (%)**
DeepPrint [69]	4.07
+ Finetune	2.68
+ 512D	2.64
+ Augmentations	2.35
+ Adversarial Loss	2.08

* Each row adds on to the previous row.

Table 3.9 Multi-finger fusion verification results on the ZJU dataset

Finger Type	**EER (%)**	**TAR (%) @ FAR = 0.01%**
Thumb	0.95	95.89
Index	0.48	98.31
LT + LI	0.00	99.77
RT + RI	0.00	99.74
RT + LT	0.00	99.80
RI + LI	0.00	99.89

Table 3.10 Separate warping modules for thumbs vs. non-thumbs (TAR (%) @ FAR=0.01%)

Method	**Trained on Thumbs and Non-Thumbs**	**Trained on Non-Thumbs**	**Trained on Thumbs**
ZJU Non-Thumbs	92.22	92.24	91.66
ZJU Thumbs	80.66	77.50	76.80
ZJU All	86.42	84.80	84.46

3.4.5 Search Experiments

For the identification (or search) experiments, we utilize the first impressions of both the contactless and contact-based fingerprints of the ZJU dataset. The contact-based fingerprints are placed in the gallery which is augmented with 1.1 million fingerprint images from an operational forensic database [262]. The contactless fingerprint images serve as the probes. We note that

Table 3.11 Improvement in minutiae correspondence without and with warping correction on ZJU dataset

	Avg. Number of Paired Minutiae	**Avg. Number of Missing Minutiae**	**Avg. Number of Spurious Minutiae**	**Goodness Index** [195]
Without Warping	28.06	67.95	71.13	−0.0167
With Warping	**30.11**	**65.00**	**69.20**	**−0.0157**

our 1.1 million augmented gallery is significantly larger than any of the existing galleries used to evaluate contact-contactless fingerprint search and is more indicative of the real world use-case of cross fingerprint matching (*e.g.*, in a National ID system like Aadhaar where a large gallery of contact-based fingerprints is already enrolled and used for de-duplication).

We evaluate 3 different search algorithms on the ZJU augmented gallery: (i) Verifinger 1:N search, (ii) search via our DeepPrint texture matcher (scores s_t from Eq. 3.12 are computed between a given preprocessed, contactless probe and all 1.1 million contact-based fingerprints in the gallery), and (iii) a two-stage search algorithm [69] where the DeepPrint texture scores are first used to retrieve the top-500 candidates, followed by a reordering using the 1:1 minutiae matching scores (s_m from Eq. 3.12) from Verifinger. The advantage of the two-stage search scheme is that it balances both speed and accuracy by utilizing the matching speed of DeepPrint to locate the first list of 500 candidates and the accuracy of Verifinger to further refine this list.

From Table 3.12, we observe that Verifinger outperforms DeepPrint stand-alone but at a search time against 1.1 million that is quite slow in comparison to DeepPrint. This motivates combining both approaches into the aforementioned two-stage search algorithm which outperforms Verifinger at rank-1 and reduces the search time by 50 seconds. In short, our two stage search algorithm obtains high levels of search accuracy on a large-scale gallery at a significant search time savings.

Table 3.12 Search performance of the proposed matcher on the ZJU dataset with a gallery of 1.1 million.

Method	**Rank (%)**				**Search**
	1	**10**	**100**	**500**	**Time (s)**
DeepPrint	83.56	93.06	95.86	97.08	0.4
Verifinger 12.0	95.25	96.47	96.95	97.20	60.1
DeepPrint + Verifinger 12.0	95.49	96.10	96.95	97.08	10.5

DeepPrint + Verifinger 12.0 refers to indexing the top-500 candidates with DeepPrint and then re-sorting those 500 candidates using a fusion of the Verifinger and DeepPrint score.

3.4.6 Segmentation Evaluation

A successful segmentation algorithm for contactless fingerprint images must not only reliably detect the distal phalange of the contactless fingerprint but also be robust to varying illumination, background, and resolution that is expected to occur in highly unconstrained capture environments.

Table 3.13 Intersection Over Union (IOU) for Segmentation $S(\cdot)$

Method	**IOU**
Baseline [158]	0.747
Proposed	0.899

The method by Malhotra et al. [158] performed well on the ISDFPDv2 dataset using certain hyperparameters that were fit to this particular dataset; however, the authors did not evaluate it on unseen datasets. In contrast, our algorithm requires no hyperparameter tuning and still performs well across a variety of different evaluation datasets, both seen and unseen. Table 3.13 gives a comparison on the unseen ZJU dataset between our method and our implementation of the baseline approach of Malhotra et al., which was trained on the ISPFDv2 dataset. For this evaluation, we manually marked the first contactless fingerprint image of each unique finger in the ZJU dataset with ground truth segmentation masks of the distal phalange and then computed the Intersection Over Union (IOU) metric between the predicted segmentation masks of our algorithm and our implementation of the benchmark algorithm in [158]. Our method does not require any hyperparameter tuning and still achieves higher IOU compared to [158].

A qualitative analysis of our segmentation network (see Figure 3.5) shows our algorithm is robust to varying illumination, background, and resolution and generalizes across multiple datasets of contactless fingerprints. However, as seen in Figure 3.5 (b), the network may still fail in extremely challenging background and illumination settings. An additional consideration, which is of importance for real-time deployment, is the processing speed of the segmentation network. Our segmentation algorithm is extremely fast compared to existing methods - requiring just 12.6ms to segment a (900×1200) resolution image. In contrast, our parallel implemenation of the baseline approach of Malhotra et al. requires 3s per image.

3.5 Discussion

Despite the low error rates achieved across each dataset, there are many factors that complicate the cross-matching performance and lead to both type I (false rejects) and type II (false accepts) errors. Many of the type I and type II errors are attributed to a failure to correctly segment and scale

only the distal phalange of the input contactless fingerprint. Incorrect segmentation can lead the large amounts of the image containing background rather than the relevant fingerprint region. Other errors can be attributed to the inherent low-contrast of the contactless fingerprints, despite any effort of contrast or resolution enhancement. The only way to mitigate these types of failures is to include a quality assurance algorithm at the point of capture of the contactless fingerprint images. Lastly, minimal overlap in the fingerprint ridge structure between genuine probe and gallery fingerprint images is the cause of many false rejections, whereas very similar ridge structure between imposter fingerprint pairs leads to a number of false accepts. This challenge is present in contact-contact matching; however, it is exaggerated in C2CL because of the unconstrained pose variance of the finger in 3D space.

The potential for greater variance in the capture conditions when capturing contactless fingerprint images necessitates more robust preprocessing to reliably match contactless fingerprints. Thus, performance will likely be markedly lower in unconstrained scenarios compared to highly controlled capture environments that employ dedicated hardware for the image acquisition, such as the PolyU and UWA datasets. However, C2CL has pushed the SOTA forward both in matching more unconstrained fingerphotos and the more constrained dedicated-device captured contactless fingerprints. Additionally, one might consider acquiring multiple image views of the same finger to build a complete 3D model of the finger to guide the preprocessing stage; however, this would add additional computational costs and latency to the acquisition process. Furthermore, in some capture scenarios, certainly the setup employed by our capture app, this process may be ergonomically challenging for the user.

As highlighted in the ablation study of Table 3.5, most of the improvement in interoperability between contactless and contact-based fingerprints is due to appropriate 500 ppi scaling of the contactless prints; however, incorporating a deformation correction module is also shown, with statistical significance, to further improve the compatibility. Figure 3.7 aims to highlight this

The Mann-Whitney rank test [55] was used to compute the statistical significance between the ROC curves of S+E+T_s and S+E+T_s+T_d. For all four datasets, the p value is smaller than 0.05, indicating that the difference is statistically significant to reject the hypothesis that the two curves are similar with a confidence of 95%.

Table 3.14 DeepPrint performance pre-trained with NIST N2N [78] dataset vs. Longitudinal dataset referenced in Yoon and Jain [262].

Dataset	Pretrained on NIST N2N Dataset (publicly available)		Pretrained on Longitudinal Dataset	
	EER (%)	TAR (%) @ FAR = 0.01%	EER (%)	TAR (%) @ FAR = 0.01%
PolyU	2.04	71.30	2.37	72.07
UWA	5.62	56.99	5.29	83.40
ISPFDv2	2.60	81.83	2.33	84.33
ZJU†	3.09	77.92	2.08	86.42

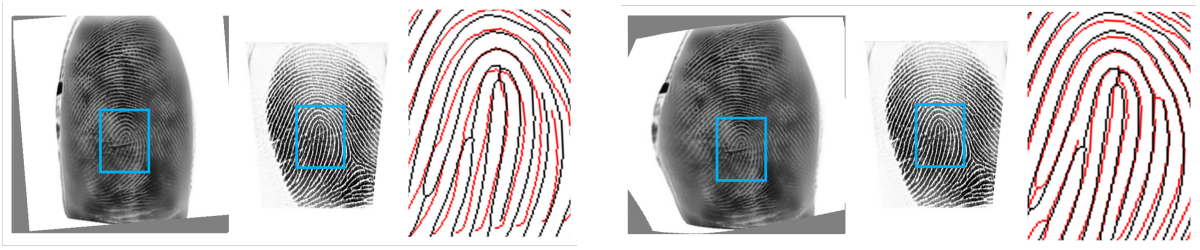
† Cross-database evaluation, *i.e.*, not seen during training.

fact through an overlay of the fingerprint ridge structure of one pair of corresponding contact and contactless fingerprints before and after applying the deformation correction. Additionally, Table 3.11 shows the average number of paired minutiae, missing minutiae, spurious minutiae, and Goodness Index [195] without and with the warping correction on the ZJU dataset. The GI, ranging from -1 to 3, is a combined measure of paired, missing, and spurious minutiae. The warping module improved the GI by 5.99%. Thus, the improved alignment indubitably leads to better minutiae-based and texture-based matching, as verified by our experiments.

Lastly, in order to utilize a large CNN, such as DeepPrint, for the task of contact-contactless fingerprint matching, we leveraged a large dataset [262] from a related domain of contact-contact fingerprint matching to pretrain our DeepPrint network. Since this dataset is not currently publicly available, we have repeated the verification experiments when pretraining DeepPrint on the publicly available NIST N2N dataset [78] (see Table 3.14). Due to the smaller dataset, we experience a slight degradation in the DeepPrint performance on some of the evaluation datasets; however, further data augmentation and incorporation of other publicly available datasets can be used to improve the performance.

3.6 Computational Efficiency

Our system architecture consists of a variety of deep networks (segmentation network, deformation correction and scaling STN, DeepPrint CNN feature extractor) and a minutiae feature extractor. The inference speeds of the segmentation network, STN, and DeepPrint are approximately 12.6ms, 6.2ms, and 26.3ms using a single NVIDIA GeForce RTX 2080 Ti GPU and 143.8ms, 19.5ms, and 120.2ms on an Intel Core i7-8700X CPU @ 3.70GHz, respectively. The Verifinger 12.0 feature



(a) (b)

Figure 3.7 Comparison of ridge overlap (a) without and (b) with the unwarping module. Use of the unwarping module results in better ridge alignment between contactless and contact-based images.

extractor requires 600ms on an Intel Core i7-8700X. In total, the inference speed of the end-to-end network is ≈ 643.6ms with an NVIDIA GeForce RTX 2080 Ti GPU or ≈ 883.5ms on an Intel Core i7-8700X CPU.

The deep network components of our algorithm are capable of very fast inference per input image; however, the system as a whole consumes a large amount of memory (400 MB). To fit into a resource constrained environment, such as a mobile phone, further optimization to the system architecture can easily be implemented with very little, if any, performance drop. First, the intermediate step of generating a scaled image prior to the deformation correction is not required for deployment and was just included for the ablation study. Instead, we can remove the affine transformation layer of our STN and directly scale and warp the input images in one step. As it stands, the main components of the algorithm, DeepPrint and Verfinger, require ≈ 1s and ≈ 1.2s on a mobile phone (Google Pixel 2), respectively. Thus, the inference time is estimated to be ≈ 2 seconds. However, to further boost the speed, rather than rely on a COTS system for minutiae extraction and matching, we can directly use the minutiae sets output by DeepPrint and a computationally efficient minutiae matcher to obtain the minutiae match scores, such as MSU's Latent AFIS Matcher [31]. Porting the model with these optimizations to a mobile phone remains as a point of future work.

3.7 Conclusion and Future Work

In this chapter, we have presented an end-to-end system for matching contactless fingerprints (*i.e.*, finger photos) to contact-based fingerprint impressions that significantly pushes the SOTA

in contact-contactless fingerprint matching closer to contact-contact fingerprint matching. In particular, our contact to contactless matcher achieves less than 1% EER across multiple datasets employing a variety of contactless and contact-based acquisition devices with varying background, illumination, and resolution settings. Critical to the success of our system is our extensive preprocessing pipeline consisting of segmentation, contrast enhancement, 500 ppi scale normalization, deformation correction, and our adaptation of DeepPrint for contact-contactless matching. Our cross-database evaluations and large-scale search experiments are more rigorous evaluations than what is reported in the open literature, and it enables us to confidently demonstrate a step toward a contact-contactless fingerprint matcher that is comparable to SOTA contact-contact fingerprint matching accuracy. The following chapter focuses on further improving sensor interoperability of fingerprint recognition systems by focusing on the diversity of fingerprint representations used for matching.

3.8 Acknowledgment

This material is based upon work supported by the Center for Identification Technology Research and the National Science Foundation under Grant No. 1841517. The authors would like to thank Debayan Deb for his help in developing the mobile phone contactless fingerprint capture application, Dr. Eryun Liu's research group at Zhejiang University for overseeing the data collection effort for this project, and the various research groups who have shared their datasets that were used in this study.

CHAPTER 4

UNIVERSAL FINGERPRINT REPRESENTATION VIA MULTIMODEL EMBEDDINGS

This chapter aims to improve the generalization capability of fingerprint recognition across a wide range of fingerprint sensors and cross-domain applications (contact to contactless, latent to rolled, etc.) by leveraging complimentary features derived from multiple state-of-the-art deep learning architectures into a single architecture. The proposed architecture, AFR-Net (Attention-Driven Fingerprint Recognition Network), outperforms several baseline models, including a SOTA commercial fingerprint system by Neurotechnology, Verifinger v12.3, across intra-sensor, cross-sensor, and latent to rolled fingerprint matching datasets. Additionally, a novel realignment strategy using local embeddings extracted from intermediate feature maps within the networks is proposed to refine the global embeddings in low certainty situations, which boosts the overall recognition accuracy. This realignment strategy requires no additional training and can be applied as a wrapper to any existing deep learning network (including attention-based, CNN-based, or both) to boost its performance in a variety of computer vision tasks.

4.1 Introduction

Automated fingerprint recognition systems have continued to permeate many facets of everyday life, appearing in many civilian and governmental applications over the last several decades [160]. As an example, India's Aadhaar civil registration system is used to authenticate approximately 70 million transactions per day, primarily with fingerprints. Due to the impressive accuracy of fingerprint recognition algorithms (0.626% False Non-Match Rate at a False Match Rate of 0.01% on the FVC-ongoing 1:1 hard benchmark [63]), researchers have turned their attention to addressing difficult edge-cases where accurate recognition remains challenging, such as partial overlap between two candidate fingerprint images and cross-sensor interoperability (e.g., optical to capacitive, contact to contactless, latent to rolled fingerprints, etc.), as well as other practical

This chapter was previously published as S. A. Grosz and A. K. Jain, "AFR-Net: Attention-Driven Fingerprint Recognition Network", IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 6, pp. 30-42, 2023.

https://uidai.gov.in/aadhaar_dashboard/auth_trend.php

Figure 4.1 Example correspondence between local features extracted from the intermediate feature maps of our AFR-Net model for two images of the same finger. Note, these local features are not necessarily the same as minutiae points, which are commonly used in fingerprint recognition.

problems like template encryption, privacy concerns, and matching latency for large-scale (gallery sizes on the order of tens or hundreds of millions) identification.

For many reasons, some of which mentioned above (e.g., template encryption and latency), methods for extracting fixed-length fingerprint embeddings using various deep learning approaches have been proposed. Some of these methods were proposed for specific fingerprint-related tasks, such as minutiae extraction [51, 234] and fingerprint indexing [27, 218], whereas others were aimed at extracting a single “global" embedding [68, 140, 143]. Of these methods, the most common architecture employed is the convolutional neural network (CNN), often utilizing domain knowledge (e.g., minutiae [68]) and other tricks (e.g., specific loss functions, such as triplet loss [61]) to improve fingerprint recognition accuracy. More recently, motivated by the success of attention-based Transformers [241] in natural language processing, the computer vision field has seen an influx of the use of the vision transformer (ViT) architecture for various computer vision tasks [38, 64, 104, 149].

In fact, two studies have already explored the use of ViT for learning discriminative fingerprint embeddings [95, 230], albeit, with the following limitations: i.) the authors of [230] supervised their ViT model using a pretrained CNN as a teacher model and thus did not give the transformer architecture the freedom to learn its own representation and ii.) the authors of [95] were limited

in the data and choice of loss function used to supervise their transformer model, thereby limiting the fingerprint recognition accuracy compared to the baseline ResNet50 model. Nonetheless, the authors in [95] did note the complimentary nature between the features learned by the CNN-based ResNet50 model and the attention-based ViT model. This motivated us to evaluate additional attention-based models that bridge the gap between purely CNN and purely attention-based models, in order to leverage the benefits of each. Toward this end, we evaluate two ViT variants (vanilla ViT [64] and Swin [149]) along with two variants of a CNN model [106] (ResNet50 and ResNet101) for fingerprint recognition. In addition, we propose our own architecture, AFR-Net (Attention-Driven Fingerprint Recognition Network), consisting of a shared feature extraction and parallel CNN and attention classification layers.

Even though these models are trained to extract a single, global embedding representing the identity of a given fingerprint image, we make the observation that for both CNN-based and attention-based models, the intermediate feature maps encode local features that are also useful for relating two candidate fingerprint images. Correspondence between these local features can be used to guide the network in placing attention on overlapping regions of the images in order to make a more accurate determination of whether the images are from the same finger. Additionally, these local features are useful in explaining the similarity between two candidate images by directly visualizing the corresponding keypoints, as shown in Figure 4.1.

One remaining concern with regards to deep learning-based fingerprint matchers is their generalization across different fingerprint sensing technology (e.g., optical, capacitive, etc.), fingerprint readers (e.g., CrossMatch, GreenBit, etc.), and fingerprint impression types (e.g., rolled, plain, contactless, etc.). This problem is often referred to as sensor interoperability, which has received some attention in recent years [4, 142, 152, 203]. In this paper, we demonstrate the generalizability of our learned representations via extensive experiments across a wide range of fingerprint sensors and types. As we show in the ablation study in section 4.4.5, much of the challenge of sensor interoperability is mitigated by training on a large, diverse training dataset; however, additional performance gains are achieved by incorporating both of the complimentary CNN and attention-based

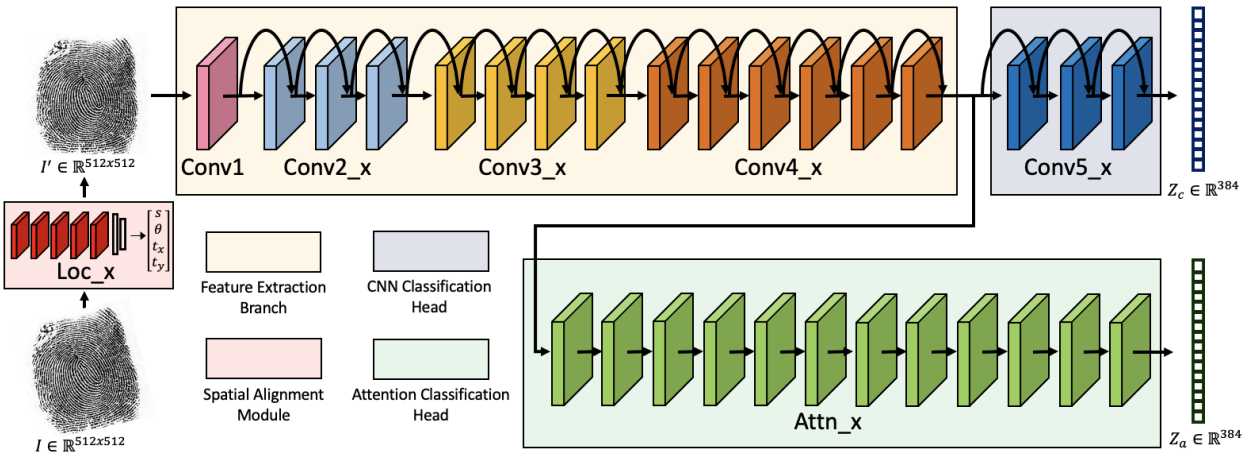


Figure 4.2 Overview of the AFR-Net architecture. First, input fingerprint images are passed through a spatial alignment module for better alignment of two fingerprints under comparison, then passed through a shared feature extraction, followed by two classification heads (one CNN-based and the other attention-based). For our implementation, we followed the ResNet50 architecture as our backbone and CNN classification head and used 12 multi-headed attention transformer encoder blocks for the attention-based classification head.

features into our network.

More concisely, the contributions of this research are as follows:

- Analysis of various attention-based architectures for fingerprint recognition.
- Novel architecture for fingerprint recognition, AFR-Net, which incorporates attention layers into the ResNet architecture.
- State-of-the-art (SOTA) fingerprint recognition performance (authentication and identification) across several diverse benchmark datasets, including intra-sensor, cross-sensor, contact to contactless, and latent to rolled fingerprint matching.
- Novel use of local embeddings extracted from intermediate feature maps to both improve the recognition accuracy and explainability of the model.
- Ablation analysis demonstrating the importance of each aspect of our model, including choice of loss function, training dataset size, use of spatial alignment module, use of both classification heads, and use of local embeddings to refine the global embeddings.

4.2 Related Work

Here we briefly discuss the prior literature in deep learning-based fingerprint recognition and the use of vision transformer models for computer vision. For a more in-depth discussion on these topics, refer to one of the many survey papers available (e.g., [172] for deep learning in biometrics and [128] for the use of transformers in vision).

4.2.1 Deep Learning for Fingerprint Recognition

Over the last decade, deep learning has seen a plethora of applications in fingerprint recognition, including minutiae extraction [51, 234], fingerprint indexing [27, 218], presentation attack detection [41, 72, 92, 240], synthetic fingerprint generation [71, 96, 129, 251], and fixed-length fingerprint embeddings for recognition [68, 140, 143]. For purposes of this paper, we limit our discussion to fixed-length (global) embeddings for fingerprint recognition.

Among the first studies on extracting global fingerprint embeddings using deep learning was proposed by Li et al. [140] which used a fully convolutional neural network to produce a final embedding of 256 dimensions. The authors of [68] then showed improved performance of their fixed-length embedding network by incorporating minutiae domain knowledge as an additional supervision. Similarly, Lin and Kumar incorporated additional fingerprint domain knowledge (minutiae and core point regions) into a multi-Siamese CNN for contact to contactless fingerprint matching [143]. More recently, [230] and [95] proposed the use of vision transformer architecture for extracting discriminative fixed-length fingerprint embeddings, both of which showed that incorporating minutiae domain knowledge into ViT improved the performance.

4.2.2 Vision Transformers for Biometric Recognition

Transformers have led to numerous applications across the computer vision field in the past couple of years since they were first introduced for computer vision applications by Doesovitskiy et al. in 2021 [64]. The general principle of transformers for computer vision is the use of the attention mechanism for aggregating sets of features across the entire image or within local neighborhoods of the image. The notion of attention was originally introduced in 2015 for sequence modeling

by Bahdanau et al. [10] and has been shown to be a useful mechanism in general for operations on a set of features. Today, numerous variants of ViT have been proposed for a wide range of computer vision tasks, including image recognition, generative modeling, multi-model tasks, video processing, low-level vision, etc. [128].

Some recent works have explored the use of transformers for biometric recognition across several modalities including face [269], finger vein [115], fingerprint [95, 230], ear [3], gait [54], and keystroke recognition [225]. In this work, we improve upon these previous uses of transformers by evaluating additional attention-based architectures for extracting global fingerprint embeddings.

4.3 AFR-Net: Attention-Driven Fingerprint Recognition Network

Our approach consists of i.) investigating several baseline CNN and attention-based models for fingerprint recognition, ii.) fusing a CNN-based architecture with attention into a single model to leverage the complimentary representations of each, iii.) a strategy to use intermediate local feature maps to refine global embeddings and reduce uncertainty in challenging pairwise fingerprint comparisons, and iv.) use of spatial alignment module to improve recognition performance. Details of each component of our approach are given in the following sections.

4.3.1 Baseline Methods

First, we improve on the initial studies [95,230] applying ViT to fingerprint recognition to better establish a fair baseline performance of ViT compared to CNN-based models. This is accomplished by removing the limitations of the previous studies in terms of choice of supervision and size of training dataset used to learn the parameters of the models. We then compare ViT with two variants of the ResNet CNN-based architecture, ResNet50 and ResNet101. For our specific choice of ViT, we decided on the small version with patch size of 16, number of attention heads of 6, and layer depth of 12. We selected this architecture as it presents an adequate trade-off in speed and accuracy compared to other ViT variants. In addition, we compare the performance of a popular ViT successor, Swin, which uses a hierarchical structure and shifted windows for computing attention within local regions of the image. Specifically, we used the small Swin architecture with patch size of 4, window size of 7, and embedding dimension of 96.

For additional baseline comparisons with previous methods, we included the latest version of the commercial-of-the-shelf (COTS) fingerprint recognition system from Neurotechnology, Verifinger v12.3, and DeepPrint [68], a fingerprint recognition network based on Inceptionv4 backbone that incorporates fingerprint domain knowledge into the learning framework. According to the FVC On-going competition, Verifinger is the top performing algorithm for the 1:1 fingerprint verification benchmark [63], and DeepPrint has also shown competitive performance with Verifinger on some benchmark datasets [68].

4.3.2 Proposed AFR-Net Architecture

Based on previous research suggesting the complimentary nature of ViT and ResNet embeddings, we were motivated to merge the two into a single architecture, referred to as AFR-Net. As shown in Figure 4.2, AFR-Net consists of a spatial alignment module, shared CNN feature encoder, CNN classification head, and an attention classification head. The shared alignment module and feature encoder greatly reduces the number of parameters compared to the fusion of the two separate networks and also allows the two classification heads to be trained jointly.

Due to the two classification heads, we have two bottleneck classification layers which map each of the 384-d embeddings, Z_c and Z_a, into a softmax output representing the probability of a sample belonging to one of N classes (identities) in our training dataset. We employ the Additive Angular Margin (ArcFace) loss function to encourage intra-class compactness and inter-class discrepancy of the embeddings of each branch [56]. Through an ablation study, presented in section 4.4.5, we find that despite the relatively little use of this loss function in previous fingerprint recognition papers [95, 187], the ArcFace loss function makes an enormous difference in the performance of our model.

4.3.3 Global Embedding Refinement via Local Embeddings

As noted in the introduction, and demonstrated in Figure 4.1, we find that the intermediate feature maps of our AFR-Net model (and in general, other deep learning models evaluated in this work) encode local descriptors (i.e., embeddings) of the input images. These local descriptors can

https://neurotechnology.com/verifinger.html

be matched between two fingerprint images and used to compute a correspondence between similar regions. Given the high accuracy of these local embeddings in locating corresponding points of interest between two images, we devise a strategy to use these corresponding regions of interest as a sort of hard attention for the model to refine the global embeddings based on just the overlapping regions present in both images.

Some examples of this process are demonstrated in Figure 4.3, where the correspondence between local embeddings is used to compute an affine transformation between the image pairs. Then, the non-overlapping fingerprint regions are masked and each image is presented to the network for a second time to yield a new set of embeddings. Finally, a second similarity score between the masked images is computed via a cosine similarity between the new embeddings. The similarity between the masked regions is combined via a weighted sum with the similarity score obtained from the original images to obtain a final similarity score.

For ResNet50, ResNet101, and AFR-Net, we take the last output of the Conv4 layer as our local embeddings, which has dimensions of 14x14x1024. For ViT and Swin, we take the final patch embeddings at the output of the last attention layer as the local embeddings, which has dimensions of 14x14x384. In all cases, each of these 196 local descriptors corresponds to a single 16x16 patch of the input fingerprint image. We assign the center of each patch as the keypoint associated with the corresponding local embedding when computing the correspondence points between two fingerprint images.

Indeed, computing the correspondence between sets of local descriptors of two images is time consuming, especially in computing a brute force exhaustive search to establish a 1:1 correspondence between matched descriptors. For this reason, we only employ the re-weighting strategy in low certainty scenarios (when the similarity score is close to the match threshold) to keep the amortized latency of our algorithm approximately the same as without the re-weighting process; that is, we only utilize the local descriptors if the similarity score between the original global embeddings falls between a specified range $[s_l, s_h]$. Values of 0.3 and 0.6 for s_l and s_h, respectively, were empirically determined on our validation dataset to work well across all models. Furthermore, if a

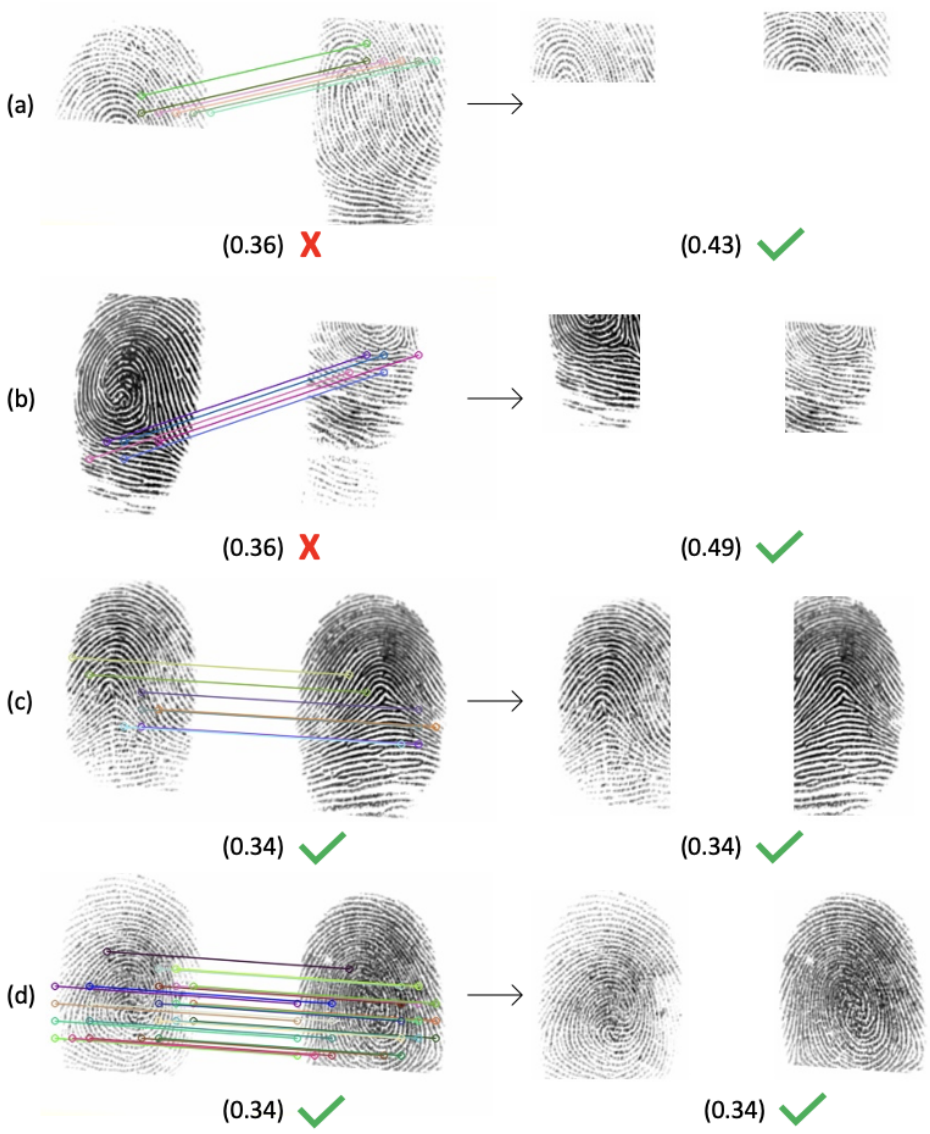


Figure 4.3 Example genuine (a and b) and imposter (c and d) pairs from FVC 2002 DB1A before and after realignment, with corresponding similarity scores from AFR-Net. Both genuine scores are pushed above the FAR=0.1% threshold of 0.36, whereas both imposter scores remained below the threshold.

Algorithm 4.1 Compute similarity between input fingerprint pairs with AFR-Net.

procedure MATCH(I_1, I_2)
  $w_1 := 0.2$
  $w_2 \leftarrow 1 - w_1$
  $w_3 := 0.5$
  $w_4 \leftarrow 1 - w_3$
  $s_l, s_h := [0.3, 0.6]$

  $Z_c^1, Z_a^1, L_1 \leftarrow AFRnet(I_1)$
  $Z_c^2, Z_a^2, L_2 \leftarrow AFRnet(I_2)$

  $s \leftarrow w_1(Z_{c1} \cdot Z_{c2}) + w_2(Z_{a1} \cdot Z_{a2})$

  if $s_l \leq s \leq s_h$ **then**
    $kp1, kp2 \leftarrow getCorr(I_1, I_2, L_1, L_2)$
    $M \leftarrow getHomography(kp1, kp2)$
    if $homographyOK(M)$ **then**
      $I_1' \leftarrow MI_1$
      $C_1, C_2 \leftarrow cropOverlap(I_1', I_2)$
      $Z_c^1, Z_a^1, \_ \leftarrow AFRnet(C_1)$
      $Z_c^2, Z_a^2, \_ \leftarrow AFRnet(C_2)$
      $s' \leftarrow w_1(Z_c^1 \cdot Z_c^2) + w_2(Z_a^1 \cdot Z_a^2)$
      $s \leftarrow w_3 s + w_4 s'$

  return s

valid homography computed between corresponding local regions cannot be obtained (e.g., if the scale, rotation, and/or translation parameters exceed expected limits), we fall back to the original similarity score as to not further degrade the comparison by computing a new set of embeddings from images which have been corrupted due to poorly behaved transformation matrices. Figure 4.4, shows the genuine and imposter score distributions for our AFR-Net model on the FVC 2002 DB3A dataset, where we experienced the biggest increase in performance after re-weighting the predictions using this method. In figure 4.4, we show (a) the original score distributions, (b) the scores computed on the refined embeddings which had well behaved homography matrices, and (c) the fused score distributions after the weighted averaging. The full algorithm of this process is detailed in Algorithm 4.1.

There are likely more sophisticated, faster alternatives to a brute force algorithm for computing

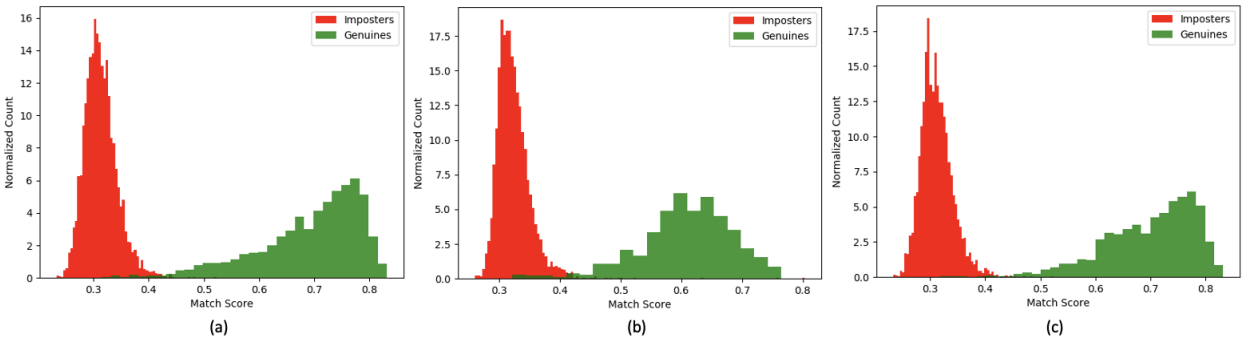


Figure 4.4 Similarity score distributions for (a) original image embeddings, (b) embeddings after refinement, and (c) weighted average of original and refined embeddings. Similarity scores are computed with the AFR-Net model on the FVC 2002 DB3A dataset, where the TAR @ 0.1% FAR for the original embeddings is 98.43%, 91.32% for the refined embeddings, and 99.36% after the weighted score fusion.

the correspondence between sets of local embeddings and aggregating the similarity scores between matched local embeddings themselves. However, we leave those areas of exploration for future work as the current algorithm seems to improve the results significantly across all models and datasets and has the nice interpretation of giving the network a second glance at regions of interest in uncertain cases - much like a human fingerprint examiner would. Nonetheless, some suggestions for future extensions would include the use of a graphical neural network (GNN) or attention mechanism to more intelligently aggregate the sets of local descriptors between two images.

4.3.4 Spatial Alignment Module

As has been noted in previous literature on fingerprint recognition [49, 68, 93, 107, 231], Spatial Transformer Networks [119] have been shown to be highly effective in aligning input fingerprints for improved recognition accuracy across a wide range of tasks (e.g., contact to contactless fingerprint matching, partial fingerprint recognition, etc.). That, coupled with our observation that the local descriptors used in our realignment procedure are not rotation invariant, we were motivated to include a spatial alignment module into the architecture of our AFR-Net model and each of the baseline models. Lastly, in the ablation portion of our experimental section, we further emphasize the benefit of incorporating the spatial alignment module into our fingerprint recognition network.

4.3.5 Training Details

AFR-Net and all baseline models, excluding COTS Verifinger and DeepPrint, were trained with an ArcFace loss function with a margin of 0.5, learning rate of 1e-4, weight decay of 2e-5, and polynomial learning rate decay function with a power of 3 and minimum learning rate of 1e-5. All models were intialized using the pre-trained ImageNet weights made available by the open-sourced pytorch-image-models git repository [249]. The AFR-Net, ResNet100, and Swin models were trained with a batch size of 64 across four Nvidia GeForce RTX 2080 Ti GPUs, whereas ResNet50 and ViT models were trained with a batch size of 128. AFR-Net, ResNet50, and ResNet100 were trained with the Adam optimizer [130] and ViT and Swin were trained with the AdamW optimizer [151]. The maximum number of epochs for all models was set to 75; however, the number of epochs trained for the final saved models varied based on the highest validation accuracy on a hold-out validation dataset.

4.4 Experimental Results

In this section, we discuss the datasets used in this study, the authentication and identification results achieved by our method in comparison with the baseline methods, latency and performance trade-off between the methods, and an ablation analysis to highlight the contributions of individual components of our algorithm.

4.4.1 Datasets

For training our models, we aggregate a large number of fingerprint datasets with diverse characteristics, ranging from rolled fingerprints [248, 262], plain (i.e., slap) fingerprints, mixture of rolled and plain fingerprints [78], contactless (e.g., from mobile phone cameras) fingerprints [17, 53, 75], latent fingerprints (from the Michigan State Police (MSP) Latent Database), and synthetic fingerprints [71, 96, 251]. Example images from each of these datasets are shown in Figure 4.5. A small portion of the total training dataset was reserved for validation. In total, our aggregated training dataset contains 1.3M images for training and 3,814 images for validation. Further information regarding the number of unique fingers, images per dataset, and fingerprint type is given in

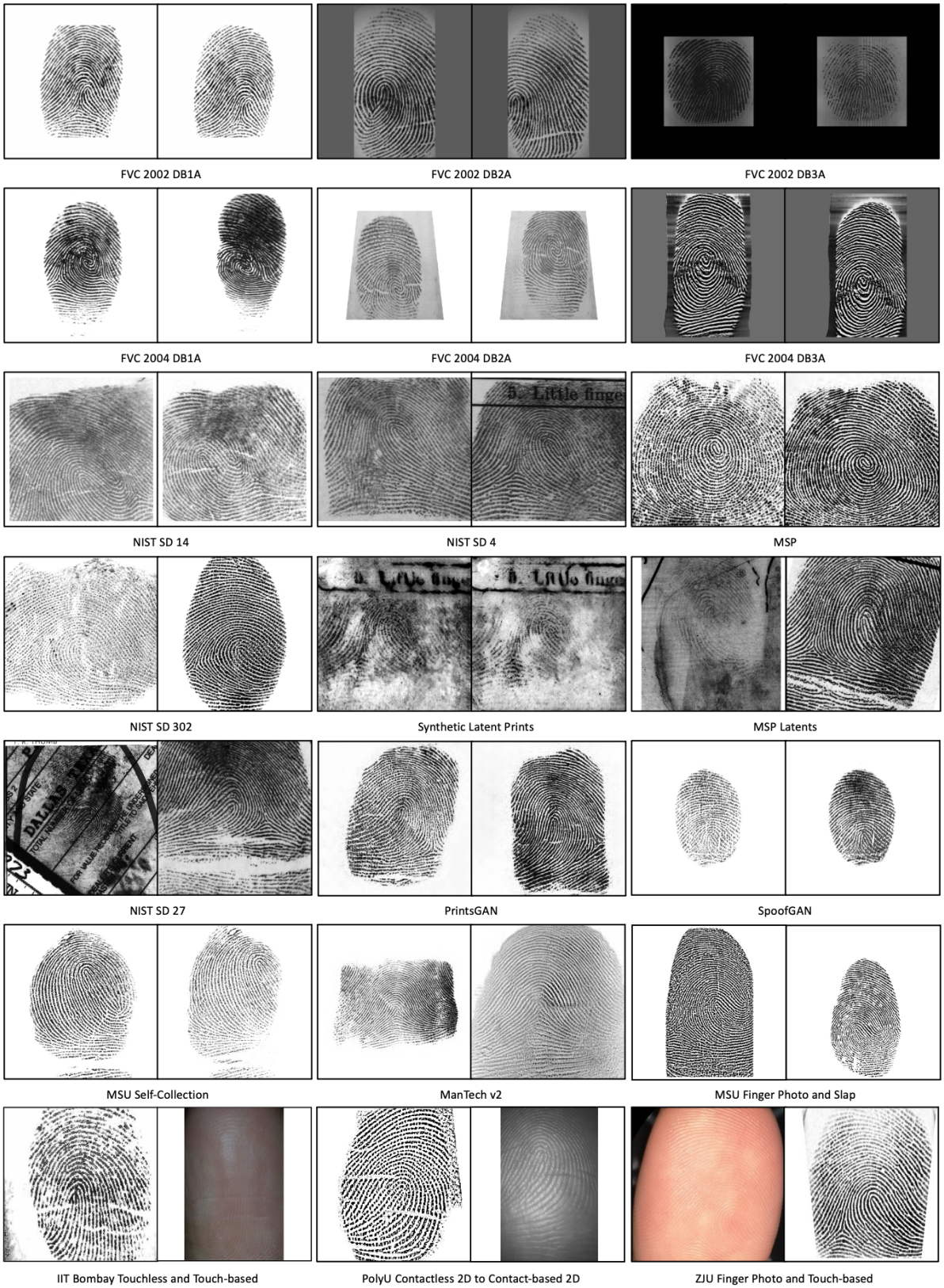


Figure 4.5 Example image pairs from each of the datasets used in this paper. See Table 7.1 for details and source of each dataset.

Table 4.1 Fingerprint datasets used in this study. Train and test splits are disjoint.

Train Dataset	**# Fingers**	**# images**
MSP† [262]	37,411	447,988
NIST SD 302 [78]	1,600	20,008
MSU Self-Collection†	4,582	57,813
PrintsGAN [71]	34,985	524,775
SpoofGAN [96]	10,000	150,000
MSU Finger Photo and Slap Database [53]	1,243	5,220
IIT Bombay Touchless and Touch-based Database [17]	200	1,600
ManTech Phase 2 [75]	4,535	64,061
Synthetic Latent Prints [251]	2,000	16,000
NIST SD 4† [248]	2,000	4,000
Validation Dataset	**# Fingers**	**# Images**
MSU Finger Photo and Slap Database [53]	110	200
MSP Latent† [262]	524	1086
NIST SD 302 [78]	200	2528
Test Dataset	**# Fingers**	**# Images**
FVC 2002 DB1A [155]	100	800
FVC 2002 DB2A [155]	100	800
FVC 2002 DB3A [155]	100	800
NIST SD 14† [246]	2700	5,400
NIST SD 302 [78]	200	2,548
NIST SD 27† [83]	258	516
PolyU Contactless 2D to Contact-based 2D Database [145]	160	960
ZJU Finger Photo and Touch-based Database [93]	824	19,776

† Not publicly available. NIST SD 4, NIST SD 14 and NIST SD 27 were publicly available but were later removed from public domain by NIST. MSP and MSP Latent databases are operational forensic datasets which cannot be released for privacy reasons and per our NDA with the Michigan State Police (MSP).

Table 7.1.

Our evaluation datasets are just as diverse as our training datasets and include challenging scenarios such as contact to contactless fingerprint matching [93, 145], varying sensor types for both rolled and slap prints (e.g., optical, capacitive, thermal swipe, etc.) [155, 246], latent to rolled fingerprint matching [83], and even rolled to plain fingerprint matching (as is the case in NIST SD 302 [78]).

4.4.2 Authentication Results

We report authentication performance of our method across 11 different evaluation datasets of varying characteristics. The results are given in Table 5.7 as the true accept rate (TAR) at a false accept rate (FAR) of 0.01% (FAR=0.1% in the case of the FVC datasets in order to follow the

We have reserved 200 of the 2,000 unique fingers in the NIST SD 302 for testing; these 200 fingers are completely disjoint from the fingers used in our training and validation partitions.

established protocols). Besides for the established protocol on the FVC datasets, we compute all possible genuine and imposter pairs for our evaluations.

According to the results in Table 5.7, AFR-Net outperforms all baseline methods on 9 out of the 11 datasets and shows competitive performance on the two datasets where it comes in second place (99.96% vs. 100% and 99.36% vs. 99.54% for FVC 2002 DB2A and DB3A, respectively). We show especially impressive performance in cross-sensor (TAR=96.11% on NIST SD 302) and contact to contactless matching (TAR=98.73% and TAR=98.70% on PolyU and ZJU datasets, respectively), as well as latent to rolled fingerprint matching on the challenging NIST SD 27 dataset, where our method out performs COTS Verifinger v12.3 (TAR=63.18% to TAR=61.63%). AFR-Net, and even our baseline ResNet and ViT variants, show substantial improvement over previous fixed-length, global representation networks for fingerprint recognition. For example, DeepPrint, one of the top performing models in the open literature, achieves a TAR of 98.55% on NIST SD 14, compared to 99.93% for AFR-Net.

For all the methods, we show improved performance with using the local embeddings to realign the images as a way to refine the global embeddings and improve the resulting similarity scores. The performance improvement was most pronounced for datasets with frequent partial fingerprints, such as FVC 2002 DB3A and DB1A. For example, the average performance across all the methods on FVC 2002 DB3A improved from 94.46% to 96.26%, a 32.5% reduction in error. Intuitively, this realignment process has the effect of slightly improving the similarity scores between borderline genuine fingerprint pairs by forcing the network to focus on overlapping regions in the images and does not appreciably effect the borderline imposter scores.

If comparing just the CNN-based models (ResNet50 and ResNet101) to the attention-based models (ViT and Swin), the performance in terms of matching accuracy is quite comparable; however, in terms of number of parameters, ViT and Swin have substantially smaller footprints. For the most part, Swin outperformed ViT in terms of accuracy across many of the datasets, but it does have more than twice the parameters and 3 times the latency of ViT, making it perhaps not as preferable in some situations.

Table 4.2 Authentication (1:1 comparison) Results.

Model	Params. (M)	Inference Speed‡ (ms)	TAR (%) @ FAR=0.1%*			TAR (%) @ FAR=0.01%				
			FVC 2002			NIST SD14	NIST SD302	NIST SD27	PolyU	ZJU
			DB1A	DB2A	DB3A					
Verifinger v12.3	N/A	600	99.96	99.86	**99.54**	98.79	93.26	61.63	95.39	96.88
DeepPrint [68]	78.81	17.53	95.32	94.64	69.93	98.55	84.01	24.81	72.07	86.42
ResNet50	62.21	4.34	99.50	99.93	93.96	**99.93**	94.70	53.88	96.94	98.28
ResNet101	81.20	7.58	99.57	99.61	94.5	**99.93**	93.79	56.59	96.48	97.71
ViT	**21.83**	**4.12**	99.29	99.68	93.00	99.67	93.49	46.51	92.38	98.08
Swin	52.69	11.66	99.75	99.79	92.43	99.89	95.46	44.96	96.15	98.33
AFR-Net	85.02	8.42	99.86	99.96	98.43	**99.93**	95.46	**63.18**	98.23	98.68
ResNet50†	62.21	10.37	99.86	**100**	95.54	**99.93**	95.40	53.88	97.60	98.04
ResNet101†	81.20	14.19	99.75	99.75	96.11	99.89	94.62	56.59	97.15	97.78
ViT†	**21.83**	10.11	99.54	99.75	95.04	99.74	94.25	46.51	94.08	98.29
Swin†	52.69	19.00	99.86	99.82	95.25	99.89	95.70	44.96	95.95	98.07
AFR-Net†	85.02	15.18	**100**	99.96	99.36	**99.93**	**96.11**	**63.18**	**98.73**	**98.70**

‡ Computed on an Nvidia GeForce RTX 2080 Ti.

* Following the FVC protocol of 2,800 genuine pairs and 4,950 imposter pairs.

† With re-weighting using local embeddings.

4.4.3 Identification Results

We used the NIST SD 27 latent fingerprint dataset and a gallery of 100K rolled fingerprints from the MSP fingerprint dataset to evaluate the closed-set identification (i.e., 1:N search) performance of our models. According to the cumulative match characteristic (CMC) curve shown in Figure 5.9 and the identification performance at specific retrieval ranks given in Table 4.3, AFR-Net is competitive with Verifinger v12.3 and outperforms all the rest of the baseline methods by a substantial margin. The rank-1 accuracy of COTS Verifinger is 55.04%, compared to 53.10% with AFR-Net but AFR-Net surpasses Verifinger at higher retrieval ranks. The next closest performing model was ResNet101, with a rank-1 accuracy of 44.96%. Some example image retrievals when (a) the correct mate was returned at rank-1 and (b) when the correct mate was not returned in the top five candidates are shown in Figure 4.6. In the successful case, the latent probe image is of relatively high quality and is able to match with its corresponding mate with a similarity score far above the other returned matches. In the failure case, the latent image is of very poor quality and returns high similarity scores with other poor quality images in the gallery.

These 100K images are completely disjoint from the 448K fingerprint images from MSP used for training.

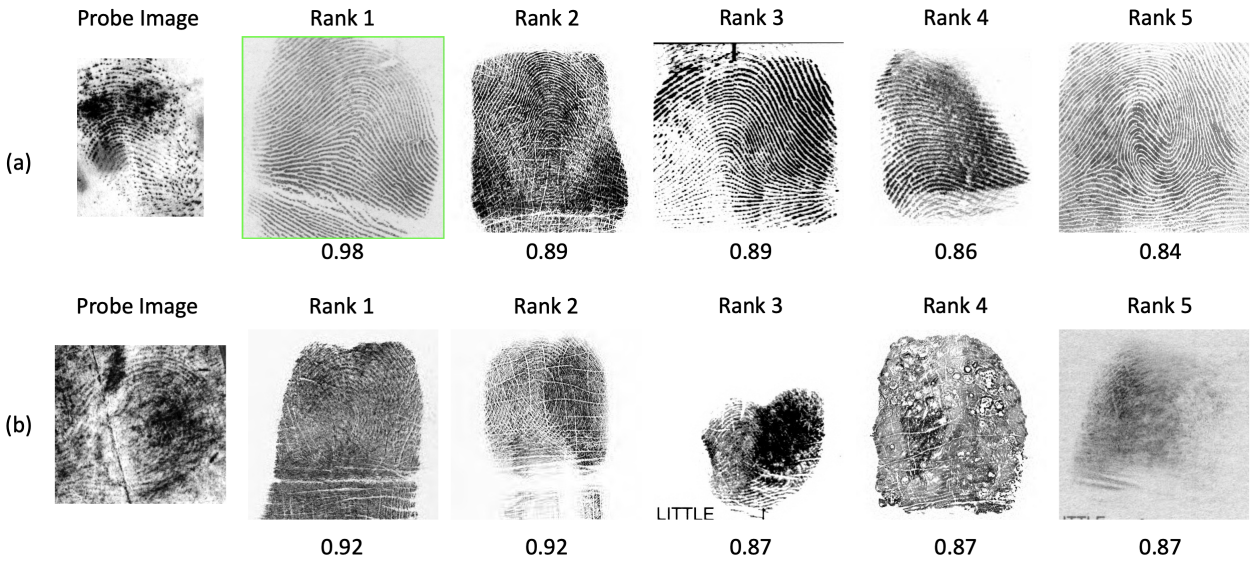


Figure 4.6 Example (a) successful and (b) unsuccessful search results for two NIST SD 27 latent probe fingerprints. In the successful case, the latent probe image is matched at rank-1 with its corresponding mate with a similarity score of 0.98; whereas, in the unsuccessful case, the poor quality latent image produces high similarity scores with other low quality images in the gallery and is not matched with the correct mate until rank 10 (with a similarity score of 0.84); not shown here since we are displaying only the top 5 retrievals.

Despite impressive performance of our model compared to the baseline methods, we should note that latent fingerprint identification is an extremely challenging task that requires targeted segmentation, enhancement, and matching strategies to achieve SOTA performance, as is demonstrated in these prior latent identification studies [28,31,121,187]. For our evaluation, we only used manual bounding box annotations to locate the latent fingerprints prior to matching, but we did not use any other preprocessing or enhancement; thus, our performance could be further improved for latent to rolled fingerprint matching. Additionally, since we do not use minutiae or any other fingerprint domain knowledge in designing AFR-Net, our model may be at a disadvantage compared to the SOTA latent matchers, since minutiae have been shown to be a useful feature for matching very low quality latents [31]. Nonetheless, AFR-Net still performs reasonably well compared to Verifinger, which is also not intended for latent to rolled fingerprint matching but does incorporate some fingerprint domain knowledge (enhancement, minutiae, etc.).

Furthermore, we observed that the fusion of Verifinger v12.3 and AFR-Net leads to a significant boost in retrieval accuracy (rank-1 accuracy of about 64% compared to 55.04% for Verifinger and

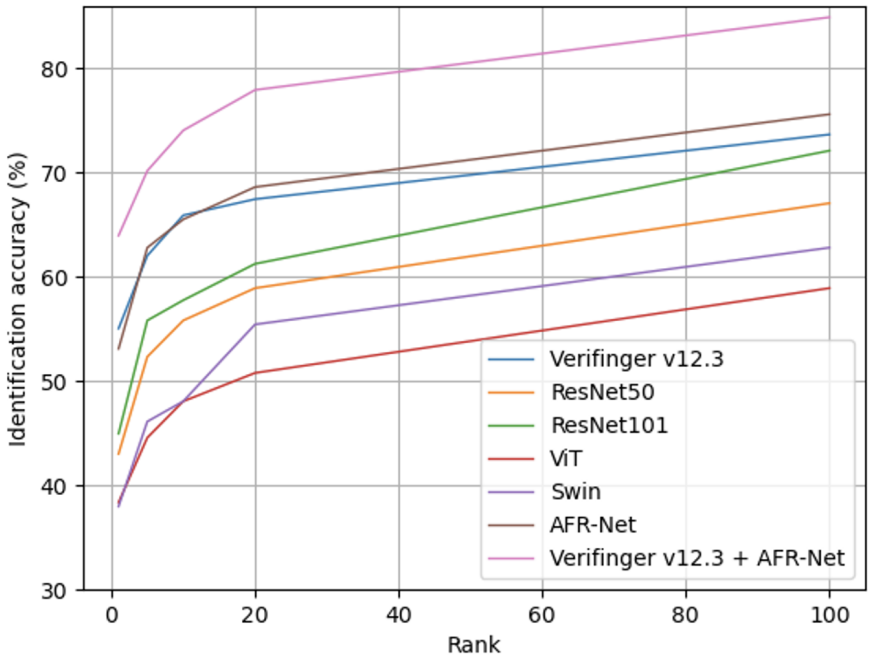


Figure 4.7 Cumulative match characteristic (CMC) curve with NIST SD 27 latent probes and a gallery of 100K rolled fingerprints plus the NIST SD 27 mated rolled fingerprint pairs.

53.10% for AFR-Net). Still, there is room for improvement, as according to Cao et al. [31], the SOTA rank-1 retrieval rate for NIST SD 27 against a gallery of 100K rolled fingerprint is 65.7%. We also evaluated the fusion of ResNet50 and ViT, which performed worse compared to using just AFR-Net (rank 1 retrieval rate of 49.61% vs. 53.10%). Thus, not only does incorporating both architectures into one save on latency and model size, as is done in AFR-Net, it also leads to better fingerprint recognition performance over the fusion of both individual models.

Lastly, we evaluated our model's performance for rolled to rolled fingerprint search using NIST SD 14. Consistent with previous studies [68], we used the last 2,700 images from NIST SD 14 as probes and their corresponding mates with the same 100K rolled images from MSP as the gallery. AFR-Net achieves a rank 1 retrieval rate of 99.78%, which is an improvement over the previous SOTA performance of 99.20% by DeepPrint [68].

4.4.4 Latency

The inference speed of each method is given in Table 5.7, along with the number of parameters of each network. Of the models that we compared, the one with the least number of parameters

Table 4.3 Closed-set identification performance on NIST SD 27 (rolled to latent comparison) at varying retrieval ranks (%) against a gallery of 100K rolled fingerprints.

Model	Rank 1	Rank 5	Rank 10	Rank 20	Rank 100
Verifinger v12.3	**55.04**	62.02	**65.89**	67.44	73.64
ResNet50	43.02	52.33	55.81	58.91	67.05
ResNet101	44.96	55.81	57.75	61.24	72.09
ViT	38.37	44.57	48.06	50.78	58.91
Swin	37.98	46.12	48.06	55.43	62.79
AFR-Net	53.10	**62.79**	65.50	**68.6**	**75.58**

is ViT (21.83 M), followed by Swin (52.69 M) and ResNet50 (62.21 M). ViT also has the lowest latency of 4.12 ms per comparison, followed closely by ResNet50 with a latency of 4.34 ms. AFR-Net has 85.02 M parameters but is still comparable to the number of parameters as ResNet101 (81.20 M).

In terms of performance vs. latency trade-off, ResNet50 outperformed ResNet101 on the majority of the evaluation datasets, whereas Swin outperformed ViT on the majority of the datasets - however, at a significant cost to latency and larger number of parameters. Thus, it seems that both ResNet50 and ViT may be preferable in some applications that require smaller footprints and faster inference speed. AFR-Net performed the best overall in terms of performance; however, AFR-Net does have a small added latency and increase in number of parameters compared to, for example, ResNet50. Still, the significant improvements in performance on many of the datasets seem to justify the added computational costs.

Lastly, the realignment stage utilizing the local embeddings does incur some additional latency, which we will denote as t_R. For our implementation, the average value of t_R is 29.36 ms. In addition to t_R, the realignment stage includes the time required for one additional inference time of the embedding network, t_I. However, since we only invoke the realignment stage for a fraction of the total comparisons, r, the amortized latency cost, t_A, of the realignment is significantly lower, computed by the following equation:

$$t_A = r(t_R + 2t_I) + (1 - r)t_I \tag{4.1}$$

For example, with a specified range of [0.3, 0.6], the realignment process for AFR-Net is invoked 17.9% (r=0.179) of the time on average across all the datasets. Using the inference speed of AFR-Net from Table 5.7 of 8.42ms, the total latency of AFR-Net† (AFR-Net with realignment) is 15.18 ms per comparison.

4.4.5 Ablation Analysis

In the abalation study of our AFR-Net model, we analyzed the effects of the loss function (cross entropy vs. ArcFace), training dataset size, use of a spatial transformer network (STN) for spatial alignment, number of classification heads, and our realignment strategy using the local feature embeddings. For the ablation on the training dataset size, we compared the performance of our algorithm when trained on only a subset of the full 1.3M training images. Specifically, we created the subset using only the publicly available fingerprint datasets, which included NIST SD 302, IIT Bombay Touchless and Touch-based, ManTech Phase 2, SpoofGAN, and PrintsGAN. This resulted in 760K training images, where 675K of these images are synthetic (from SpoofGAN and PrintsGAN). In comparison, our full training database consists of the same 675K synthetic images plus an additional 540K real fingerprint images.

The results of the ablation study are given in Table 5.9. The largest increase in performance is attributed to the use of ArcFace loss rather than a cross entropy loss for supervision. Interestingly, training with ArcFace loss on a subset of only publicly available training data (85K real fingerprints + 675K synthetic compared to our full dataset of 540K real fingerprints + 675K synthetic) achieves competitive recognition performance across all datasets, where the benefit of additional data is most evident in the cross-sensor and latent matching scenarios. Even further improvements were obtained with the incorporation of the spatial alignment network. Finally, we noticed consistent improvements across all evaluation datasets with applying our realignment strategy, especially in the more challenging datasets such as NIST SD 302 and FVC 2002 DB3A, which have many partially overlapping fingerprints.

Table 4.4 Ablation study for AFR-Net.

Loss	# Imgs.	STN	Backbone	Realign	TAR (%) @ FAR=0.1%*			TAR (%) @ FAR=0.01%				
					FVC 2002			NIST	NIST	NIST	PolyU	ZJU
					DB1A	DB2A	DB3A	SD14	SD302	SD27		
Cross Entropy	760K	No	CNN+Attn	No	96.93	96.50	80.21	98.23	74.05	21.32	69.15	87.09
ArcFace	760K	No	CNN+Attn	No	98.46	99.86	92.50	99.63	92.04	39.53	91.3	97.61
ArcFace	1.3M	No	CNN+Attn	No	99.79	99.82	97.82	99.89	94.82	58.14	97.00	98.66
ArcFace	1.3M	Yes	CNN only	No	99.64	**99.96**	96.68	99.89	95.17	55.43	96.72	98.63
ArcFace	1.3M	Yes	Attn only	No	99.86	99.93	98.34	**99.93**	95.21	60.85	98.45	98.57
ArcFace	1.3M	Yes	CNN+Attn	No	99.86	**99.96**	98.43	**99.93**	95.46	**63.18**	98.23	98.68
ArcFace	1.3M	Yes	CNN+Attn	Yes	**100**	**99.96**	**99.36**	**99.93**	**96.11**	**63.18**	**98.73**	**98.70**

* Following the FVC protocol of 2,800 genuine pairs and 4,950 imposter pairs.

4.5 Discussion

In this section, we discuss some remaining failure cases of our model and some possible future extensions to mitigate them. We also investigate the robustness of our model to partial fingerprints by manually generating affine and occlusion deformations of varying magnitudes.

4.5.1 Failure Case Analysis

Two example fingerprint image pairs that failed to be successfully matched by AFR-Net are shown in Figure 7.12. As demonstrated with these representative examples, the majority of the failure cases can be attributed to one of two factors: i.) extremely poor image quality or ii.) very little overlap between the images. The first cause can be avoided by implementing a quality check into the algorithm, whereas the second cause may be more difficult to detect and/or avoid in a practical system (especially one operating with a limited acquisition aperture). Our realignment strategy is effective at improving partial overlap pairs; however, when the amount of overlap is severe, such as example (b) in Figure 7.12, the model may still fail.

4.5.2 Robustness to Occlusions and Affine Transformations

To help understand the difference between CNN-based and attention-based embeddings, we conducted an experiment to visualize the saliency maps of each model on pairs of partial fingerprints. Specifically, we scan a mask of 16x16 pixels (with a stride of 16) across one image in the pair and compute the resulting similarity scores, which we use to draw a heatmap of salient regions for each patch in the image. We repeat this process for the other image and overlay the heatmaps onto the original images. Due to space limitations, we just show one representative example in

Figure 4.8 Example fingerprint pairs that failed to match successfully by the proposed AFR-Net. (a) is from NIST SD 302 and (b) is from FVC 2002 DB2A. Similarity scores for (a) and (b) are 0.35 and 0.34, respectively; both below the match threshold of 0.36.

Figure 4.9 along with the ridge overlays of the two images to better visualize the overlapping regions. Comparing the saliency maps of ResNet50 ((a) in Figure 4.9) to ViT ((b) in Figure 4.9), it seems that occluding some areas of the fingerprint has more of an effect on varying the similarity scores for ViT than it does for ResNet50. This suggests that ViT is placing more weight on specific regions of the fingerprint compared to ResNet50 which may be using more of the fingerprint area for its prediction. Finally, comparing both saliency maps with the saliency map of AFR-Net (shown in (c) of Figure 4.9), we can see that AFR-Net exhibits characteristics of both models.

We performed two additional experiments using manual occlusions and affine transformations to generate partial fingerprints and plotted the performance of each model vs. the amount of degradation. Specifically, we generated random occlusions and affine transformations at five different ratios, corresponding to the percentage of the fingerprint area being obscured. We repeated the experiment 5 times at each ratio and recorded the average performance of each model. For reference, example images with random affine transformations and occlusions at ratios of 40% and 20%, respectively, are shown in Figure 4.10.

According to Figure 4.11, the ResNet models appear to have a slight edge over the attention-based models (ViT and Swin) when subjected to random occlusions, whereas the ViT and Swin models show more robustness to severe degrees of affine transformations compared to the ResNet models. However, AFR-Net shows the best robustness to both occlusions and affine transformations, underscoring the benefit of merging the two complimentary networks.

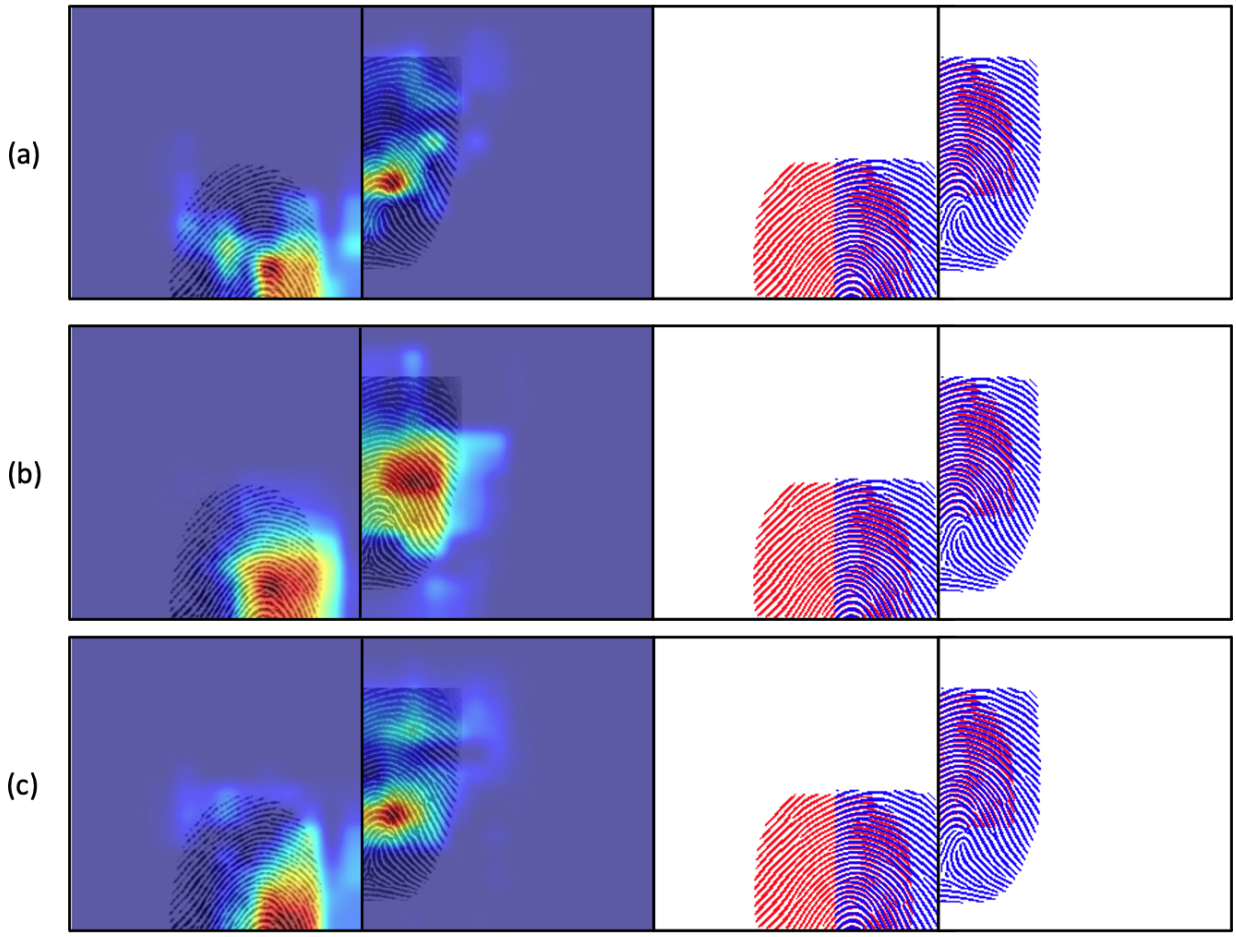


Figure 4.9 Saliency maps for a partial fingerprint pair from FVC 2002 DB1A computed with (a) ResNet50, (b) ViT, and (c) AFR-Net models. The reddish regions represent regions of the fingerprint where the similarity score drops the most when occluded (indicating it's importance), whereas the blue regions represent the regions with high similarity scores even with that region occluded (indicating low importance). On the right, the ridge structures of each fingerprint are overlaid to highlight the overlapping area. (Best viewed in color).

4.6 Conclusion

In this chapter, we evaluated attention-based fingerprint recognition networks against competitive CNN baselines and a state-of-the-art commercial fingerprint recognition system, Verifinger v12.3, and showed that our combined architecture, AFR-Net (Attention-Driven Fingerprint Recognition Network), outperforms all of the baselines in the majority of the evaluation datasets. These evaluations included challenging intra-sensor, cross-sensor, contact to contactless, and latent fingerprint matching scenarios. Furthermore, we introduced a realignment stage using the

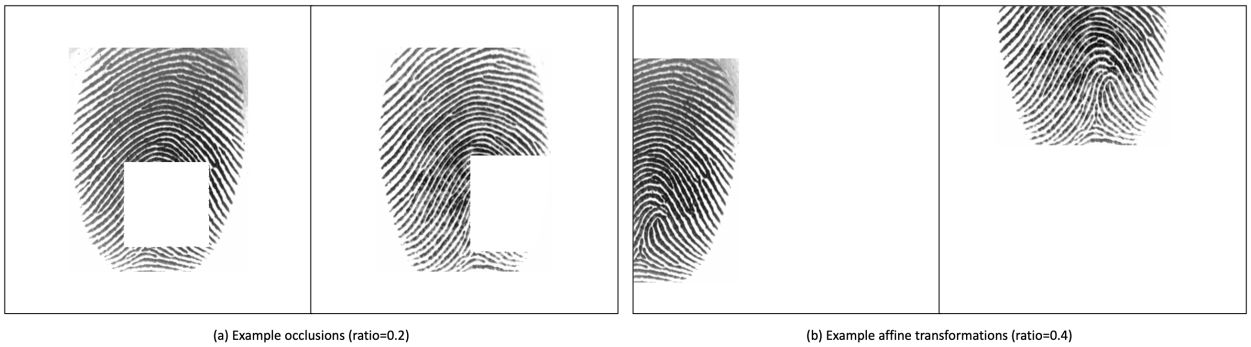


Figure 4.10 Example manual occlusions and affine transformations to generate challenging, partial fingerprint pairs.

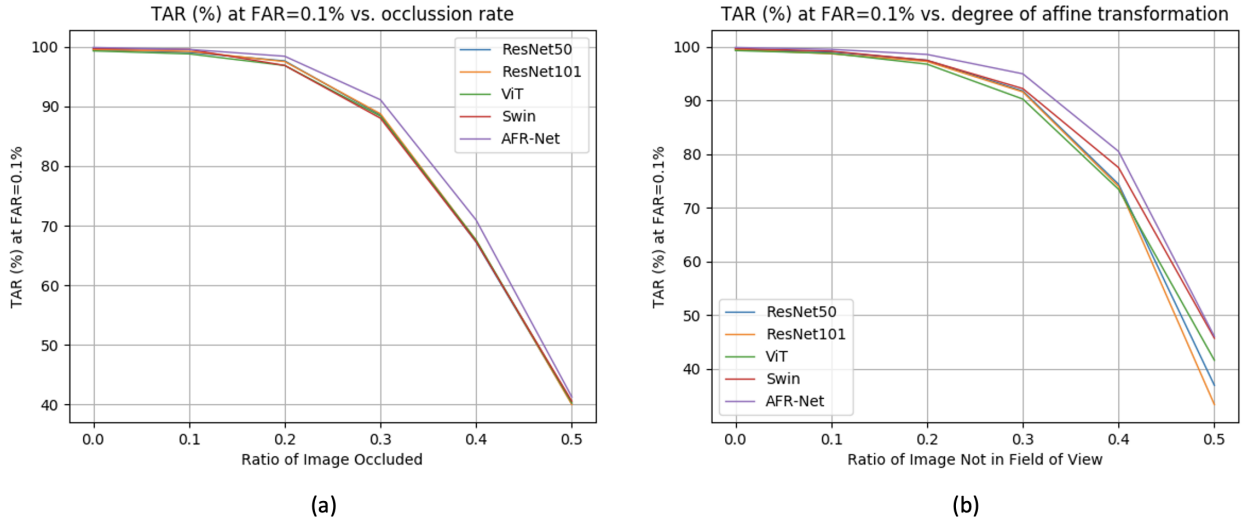


Figure 4.11 Comparison in TAR (%) at FAR=0.1% of each model under increasing degrees of random (a) occlusion and (b) affine transformations.

correspondence between local embeddings extracted from intermediate feature maps of two fingerprint images which consistently improved the performance across all the models, especially in challenging cases (e.g., partial overlap between the fingerprint images). This realignment strategy requires no additional training and can be applied as a wrapper to any deep learning network (CNN or attention-based). It also serves as an explainable visualization of the corresponding regions of two fingerprint images as ascertained by the network. Future work will aim at improving the realignment strategy to reduce the latency introduced by the current brute force correspondence implementation. The use of attention and/or graphical neural networks for this purpose may be explored in order to more intelligently aggregate two sets of local embeddings. In the next chapter,

we turn our attention to learning a diversity of scales (both global and local) encoded into the embeddings used for fingerprint recognition with a particular emphasis on improving latent to rolled fingerprint matching, one of the most challenging, unconstrained applications.

CHAPTER 5

LATENT FINGERPRINT RECOGNITION: FUSION OF LOCAL AND GLOBAL EMBEDDINGS

One of the most challenging problems in fingerprint recognition continues to be establishing the identity of a suspect associated with partial and smudgy fingerprints left at a crime scene (i.e., latent prints or fingermarks). Despite the success of fixed-length embeddings for rolled and slap fingerprint recognition, the features learned for latent fingerprint matching have mostly been limited to local minutiae-based embeddings and have not directly leveraged global representations for matching. In this chapter, we combine global embeddings with local embeddings for state-of-the-art latent to rolled matching accuracy with high throughput. The combination of both local and global representations leads to improved recognition accuracy across NIST SD 27, NIST SD 302, MSP, MOLF DB1/DB4, and MOLF DB2/DB4 latent fingerprint datasets for both closed-set (84.11%, 54.36%, 84.35%, 70.43%, 62.86% rank-1 retrieval rate, respectively) and open-set (0.50, 0.74, 0.44, 0.60, 0.68 FNIR at FPIR=0.02, respectively) identification scenarios on a gallery of 100K rolled fingerprints. Not only do we fuse the complimentary representations, we also use the local features to guide the global representations to focus on discriminatory regions in two fingerprint images to be compared. This leads to a multi-stage matching paradigm in which subsets of the retrieved candidate lists for each probe image are passed to subsequent stages for further processing, resulting in a considerable reduction in latency (requiring just 0.068 ms per latent to rolled comparison on an AMD EPYC 7543 32-Core Processor, roughly 15K comparisons per second). Finally, we show the generalizability of the fused representations for improving authentication accuracy across several rolled, plain, and contactless fingerprint datasets.

5.1 Introduction

Latent fingerprints are fingerprint impressions that are left behind, unintentionally, on surfaces such as glass, metal, and plastic, and are often invisible to the human eye. However, these prints can

This chapter was previously published as S. A. Grosz and A. K. Jain, "Latent Fingerprint Recognition: Fusion of Local and Global Embeddings", IEEE Transactions on Information Forensics and Security, vol. 18, pp. 5691-5705, 2023.

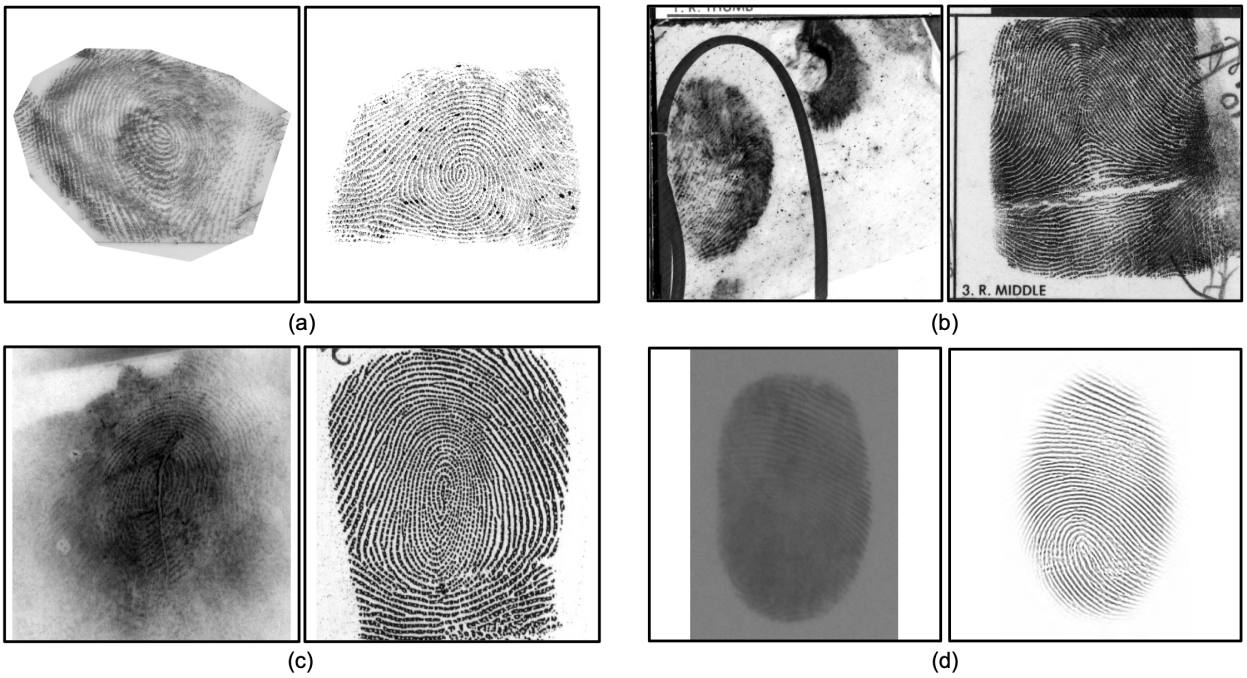


Figure 5.1 Example latent and corresponding rolled images from (a) N2N Latent [78], (b) NIST SD 27 [83], (c) MSP Latent [262], and (d) MOLF [212] datasets. In all the above examples, the true mate for each query latent was returned at rank-1 by the proposed method against a gallery of 100K rolled fingerprints.

be lifted and analyzed using specialized techniques to enhance the friction ridge patterns present in the prints [256]. Once the prints have been enhanced, digital images can be compared against a database of known ten-print fingerprints (rolled or slap) in law enforcement databases, which can help identify the person who left the prints behind. In fact, latent fingerprint recognition has been used as a tool in forensics and criminal investigations over the last century and are often regarded as a credible source of information in identifying suspects [160]; however, the reliability of automatic latent to rolled fingerprint matching considerably lags that of rolled to rolled fingerprint matching. As a result, some innocent individuals, like in the case of Brandon Mayfield [182], have unfortunately been incarcerated due to inaccurate latent to rolled comparison by automatic fingerprint identification systems (AFIS) and failure of forensic examiners to follow the ACE-V protocol, established in the 1980s [8]. As demonstrated in the four example latent fingerprints shown in Figure 5.1, some of the reasons for low performance in latent fingerprint recognition include poor ridge-valley contrast, occlusion, distortion, varying background, and incomplete

fingerprint patterns. Because of these challenges, latent fingerprint recognition remains one of the most challenging problems in biometrics, akin to matching poor quality face images from CCTV surveillance frames to mugshot photos.

In general, AFIS optimized for rolled or plain fingerprint impressions do not achieve comparable accuracy for latent fingerprints, even when finetuned on publicly available latent fingerprint datasets. For example, the commercial software Verifinger v12.3 achieves a true accept rate (TAR) of 99.93% at a false accept rate (FAR) of 0.01% on the NIST SD 14 rolled fingerprint dataset, but only a TAR of 55.04% on NIST SD 27 latent dataset. This has sparked a number of studies that have focused on improving individual components of the fingerprint recognition pipeline to work better for latent impressions, such as those focusing on foreground segmentation of the ridge structure [30], latent enhancement [116, 126, 137, 271], minutiae extraction [29, 51, 187, 233], and orientation field estimation [77, 258]. However, few studies have focused on an end-to-end system to improve the latent to rolled fingerprint recognition pipeline, which is necessary since optimizing individual components separately may lead to sub-optimal performance when integrated together and tested as a complete system. Of those studies that do report on an end-to-end recognition system [28,31,234], the highest rank-1 retrieval rate achieved is 65.7% [31], computed on 258 latent probes from NIST SD 27 against a background of 100K rolled fingerprints.

Furthermore, despite recent advancements in deep learning techniques for fixed-length representations (embeddings) for rolled/plain fingerprint recognition, these global representations have yet to be leveraged effectively for latents, likely due to the domain gap and limited availability of large-scale latent fingerprint datasets to learn such representations. Therefore, in this study, we propose an end-to-end pipeline for latent fingerprint recognition which leverages both a learned, global fingerprint representation (i.e., entire friction ridge pattern) and local representations (i.e., minutiae and virtual minutiae) for improved accuracy and search speed of latent to rolled fingerprint recognition. We not only fuse these complimentary local and global embeddings but utilize the local features to inform or guide the global representations to focus on discriminatory regions of

Virtual minutiae are densely sampled points on an evenly spaced grid on the extracted fingerprint ridge area.

the input fingerprint image pairs for an improved matching accuracy.

Lastly, existing latent to rolled fingerprint matching pipelines are highly tuned specifically for latent fingerprints, whereas our representation and matching pipeline is generalizable and effective across a wide range of fingerprint sensors (e.g., optical, capacitive, etc.) and image domains (e.g., latent, rolled, plain, contactless captures via mobile phone cameras, etc.). In a sense, we report on a *universal fingerprint representation* that is agnostic to fingerprint sensors and fingerprint capture mode. Concretely, the contributions of this research are the following:

1. Design of an end-to-end latent fingerprint recognition pipeline using deep learning methods, including algorithms for segmentation, enhancement, minutiae extraction, and a fusion of global and local embeddings.
2. State-of-the-art (SOTA) latent to rolled/plain fingerprint search across multiple datasets, including NIST SD 27 [83], NIST SD 302 Latents (N2N Latents) [78], MSP Latent [262], and MOLF datasets [212].
3. Faster search speed (low latency) due to our multi-stage search scheme while maintaining SOTA recognition accuracy for both closed-set and open-set identification.
4. Generalization of representation (embedding) from LFR-Net is shown via SOTA authentication performance across several rolled (NIST SD 14 [246]), plain (NIST SD 302 [78]), and contact to contactless fingerprint matching datasets (PolyU Contactless 2D to Contact-based 2D [145] and ZJU Finger Photo and Touch-based [93]) using the same network, a step toward a universal fingerprint recognition system.

5.2 Related Work

5.2.1 Latent Fingerprint Enhancement

A critical step in improving the accuracy of latent to rolled comparison is alleviating the effect of various degradations present in latent fingerprints through preprocessing aimed at enhancing the contrast of the latent fingerprint ridge structure. A multitude of latent enhancement methods have been proposed over the years, ranging from classical computer vision techniques [37, 40, 77, 258,

Table 5.1 Summary of recently published latent fingerprint recognition studies.

Study	Approach	Database	Rank-1
FingerNet, 2017 [234]	CNN methods for minutiae extraction, orientation field estimation, segmentation, and enhancement. Search results on a gallery of 40K.	NIST SD 27 [83]	~35.0%
MSU-AFIS, 2019 [31]	CNN methods for enhancement, segmentation, minutiae, and virtual minutiae extraction. Search results on a gallery of 100K.	NIST SD 27 [83] MSP Latent† [262]	65.7% 69.4%
Gu et al., 2020 [102]	Registration of latents via matching dense, undirected sampling points + virtual minutiae (not a fully automated system). Search results on a gallery of 100K.	NIST SD 27 [83] MOLF DB3/DB4 [212]	70.1% 19.8%
MinNet, 2022 [187]	CNN-based minutiae patch embedding network + local similarity assignment (LSA) algorithm for matching. Search results on a gallery of 5,560, 316, and 168 images from EGM, FVC-Latent, and Tshingua-Latent databases, respectively.	EGM (private dataset) FVC-Latent [187] Tshingua-Latent [187]	92.39% 95.57% 99.40%
FingerGAN, 2023 [271]	GAN-based enhancement + Verifinger v12.1. Search results on a gallery of 27,258.	NIST SD 27 [83] MOLF DB1/DB4 [212] MOLF DB2/DB4 [212] MOLF DB3/DB4 [212]	59.69% 25.34% 22.23% 29.43%
LFR-Net (proposed approach)	CNN-based latent enhancement, segmentation, and fusion of local (minutiae + virtual minutiae) and global (AFR-Net [97]) embeddings for matching. Search results reported on a gallery of 100K.	NIST SD 27 [83] N2N Latent [78] MSP Latent [262] MOLF DB1/DB4 [212] MOLF DB2/DB4 [212]	84.11% 54.36% 84.35% 70.43% 62.86%

† Did not specify the test split.

260, 264, 265] to state of the art deep learning methods [24, 48, 116, 126, 137, 147, 227, 234, 271].

Early enhancement efforts utilized contextual filtering and directional filtering [37,40], but these methods were limited in their effectiveness for enhancing latent fingerprints due to corrupted ridge structures and unreliable orientation and frequency estimation compared to that of plain and rolled fingerprints. This led to many subsequent studies on improving the ridge orientation estimation for latent fingerprints. For example, Yoon et al. [260] utilized a combination of polynomial models and Gabor filters to improve latent orientation estimation. Similarly, Feng et al. [77] utilized an orientation patch dictionary and Gabor filters for latent enhancement, and Yang et al. [258] extended this approach by utilizing local orientation dictionaries, which increased the flexibility of the approach to find better orientation fields. However, the variance in ridge frequency of distorted

latent fingerprints limited the utility of these methods in improving overall matching accuracy.

Subsequent efforts introduced deep neural networks to improve the enhancement of latent fingerprints. In addition to a combination of short-time Fourier transform (STFT) and Gabor filters, Cao et al. [24] trained a convolutional neural network (CNN) autoencoder to enhance latent fingerprints. Variants of the CNN-based approach were also proposed in [137,147,227]. Generative adversarial networks (GANs) have also been adopted for latent fingerprint enhancement, and these methods have shown promise in restoring ridge and valley structures [48, 116, 126, 271]. However, as shown in Table 5.8, these methods have a tendency to hallucinate ridge lines and produce spurious minutiae that may degrade matching performance. Furthermore, critical to the success of many of these methods was access to large databases of mated rolled and latent fingerprint image pairs for training, many of which are unfortunately not publicly available to other researchers. In this work, we adopt the efficient CNN architecture of Squeeze U-Net [14] for latent enhancement without access to any latent training data. Instead, we employ a series of data augmentations on a dataset of rolled and plain fingerprint impressions in order to mimic the degradations present in latent fingerprints, and our network is trained to restore the degraded images to their original input. A comparison between the performance of our enhancement network and several previous baselines is given in section 5.5.2.

5.2.2 Latent Fingerprint Recognition

Despite recent success of deep learning global representations for fingerprint matching, all existing latent fingerprint recognition systems (to the best of our knowledge) utilize minutiae-based matchers for computing final similarity scores between latent and rolled image pairs. For example, Cao et al. [31] and Öztürk et al. [187] utilize variants of the local similarity assignment algorithm proposed in [34] for computing minutiae similarity scores; Tang et al. [234] utilized the extended clique model for minutiae matching, FingerGAN used Verifinger v12.1 for matching; and Gu et al. [103] utilized multi-scale fixed-length embeddings for indexing to reduce the potential candidate list in combination with MSU-AFIS [31] for computing the similarity scores. Even though deep learning networks are used within many of these minutiae-based methods to produce

local minutiae descriptors around minutiae points, no existing study directly leveraged a global embedding as an additional similarity comparison. In this paper, we propose to use a global embedding score for improving the latent to rolled matching performance in conjunction with local minutiae embeddings for minutiae matching. A table giving a brief summary of the recent publications on latent fingerprint recognition is given in Table 5.1

5.3 LFR-Net: Latent Fingerprint Recognition Network

Our approach for accurate and efficient latent fingerprint search consists of a combination of local (minutiae and virtual minutiae) and global features (AFR-Net embeddings [97]). Additionally, due to the low contrast, occlusion, and varying background present in many latent fingerprint images, we first incorporate automatic segmentation and enhancement of latent fingerprint images prior to feature extraction. The following sections will describe each component of our latent to rolled fingerprint matcher, referred to as LFR-Net. These components include enhancement, segmentation, minutiae extraction, virtual minutiae extraction, global embedding, realignment for improved global embeddings, and a multi-stage search strategy. An overview of the pipeline is illustrated in Figure 5.2.

5.3.1 Latent Enhancement and Segmentation

The terminology introduced for NIST SD 27 denotes the quality of latent fingerprints as either good, bad, or ugly depending on several factors, including the percentage of the fingerprint ridge structure occluded, noise obscuring the ridge structure, and low contrast of the ridges compared to the background content of the image. To make things even more challenging, the quality and appearance of latent fingerprints can vary drastically across different databases, either collected in the lab (as is the case for the NIST SD 302 (N2N) [78] and IIIT-D MOLF datasets [212]) or from real crime scenes (as is the case for NIST SD 27 [83] and MSP Latent datasets [262]). Therefore, latent enhancement is a critical yet challenging step for accurate and reliable latent to rolled fingerprint matching. See Figure 5.3 for example latent and rolled/plain fingerprint pairs showcasing the various differences between latent datasets.

To address the problem of latent enhancement, we focus on two key factors degrading the quality

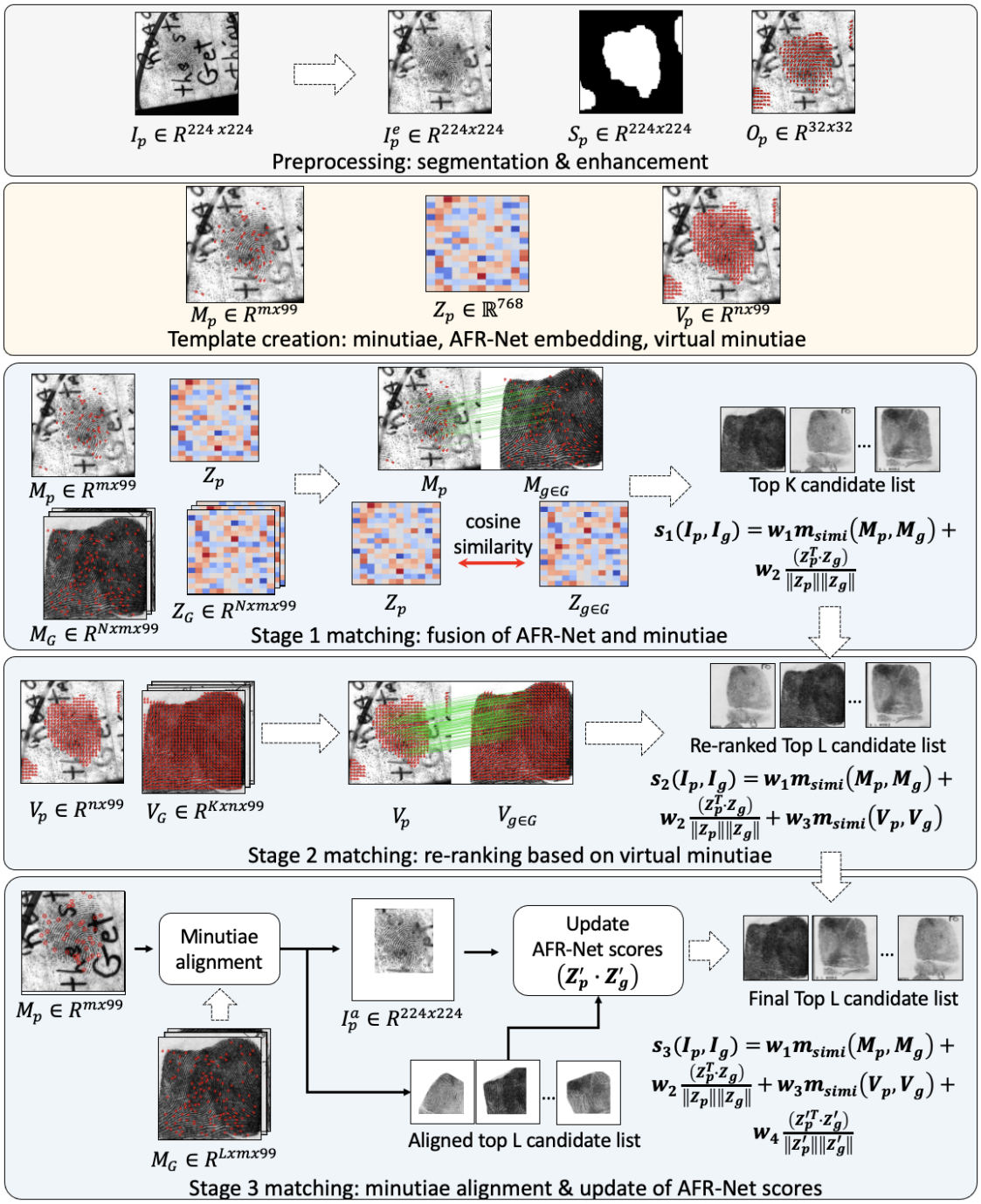


Figure 5.2 Overview of LFR-Net. An input latent probe image I_p is first automatically segmented and enhanced to generate I_p^e, orientation field O_p, and segmentation mask S_p. Then, I_p^e is passed to a minutiae extraction network, minutiae descriptor network, and AFR-Net [97] to produce a minutiae feature set M_p, virtual minutiae feature set V_p, and AFR-Net embeddings Z_p, which are embedded into a template for matching (M_p, Z_p, V_p). Once extracted, the probe feature template is compared with each gallery template (M_g, Z_g, V_g) in the gallery G of size N via a similarity function $s(I_p, I_g)$ in three stages.

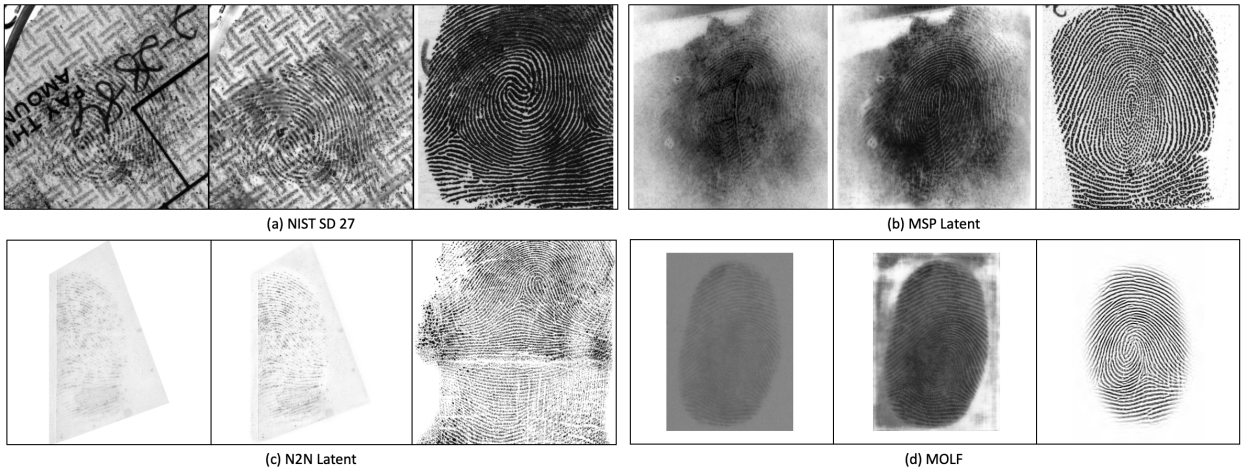


Figure 5.3 Example enhanced latent images from (a) NIST SD 27 [83], (b) MSP Latent dataset [262], (c) N2N Latent [78], and (d) MOLF [212] datasets. In each subfigure, the left image is the original latent image, the middle image is the enhanced latent image using the proposed enhancement network, and the right image is the corresponding rolled mate.

of latent prints - namely, presence of noise occluding areas of the latent fingerprint ridge structure and low contrast of the ridges. To remove noise from the latent images, we train a de-noising CNN network to remove noise and fill-in occluded regions of the fingerprint ridge structure. This network architecture is modeled from Squeeze U-Net [14], an efficient network proposed for image segmentation, where we have adapted it for latent enhancement. Next, we aim to highlight the ridge structure of the latent fingerprints by constraining the network to segment the fingerprint ridge lines from the background. To accomplish this, we introduce an additional channel to the output of our enhancement network and optimize for both tasks in a single architecture. Thus, the output of the enhancement network is two channels, one for the enhanced image and another for the ridge lines. Note, the outputs of both channels are gray-scale and in the range [0,255]. A few example enhancement outputs from this network are shown in the middle column of each sub-figure in Figure 5.3 and the bottom two rows of Figure 5.5.

To locate and segment the latent fingerprint area from the background image content, we use the predicted fingerprint ridges as a segmentation mask for localizing the latent fingerprint area by performing a series of simple image processing operations. First, a Gaussian filter with kernel size (5,5) is applied to the predicted ridge map, followed by a thresholding operation with a threshold

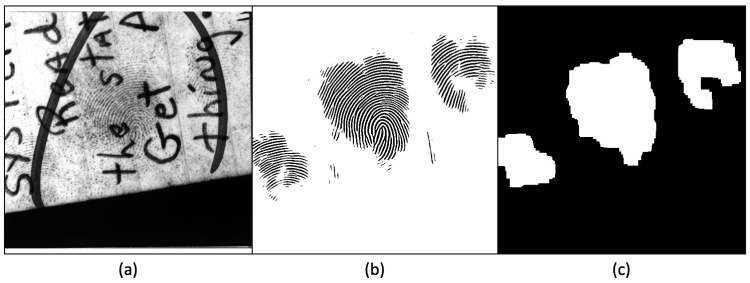


Figure 5.4 Example mask prediction for a latent image from NIST SD 27. (a) input latent image, (b) gray-scale ridge image output by the enhancement network, and (c) binary mask obtained after a series of Guassian blurring, thresholding, and morphological operations on (b).

of 150 on the pixel values to obtain the binary ridge lines in the range [0,1]. Next, a morphological closing operation with a kernel size of (9,9) is repeated 3 times, followed by three morphological opening operations with a kernel size of (9,9). Finally, the mitigate erroneous predictions, the resulting mask defaults to the entire image if the resulting mask after processing has an area of less than 10,000 pixels. Since our enhancement network is fully convolutional, it can accept images of any resolution; however, the final segmented images are cropped to a height and width of 512×512 pixels at a resolution of 500 ppi. Figure 5.4 illustrates the process of converting a predicted gray-scale ridge image to a binary segmentation mask for an example latent fingerprint from NIST SD 27.

Due to a lack of publicly available large-scale latent databases, we utilize several data augmentations to mimic the distribution of latent fingerprints using a collection of rolled and slap fingerprints. These data augmentations are illustrated in (b) of Figure 5.5 and consist of random amounts of Gaussian blurring, Gaussian noise, downsampling, partial occlusions, and contrast adjustments. The enhancement network is trained to remove these degradations from the augmented images via an MSE loss between the predicted, enhanced image and the original, unperturbed image. Furthermore, we compute an additional MSE loss between the predicted ridge images and the ridge images extracted from the original input fingerprints via Verifinger v12.3 (normalized to the range [0,255]). Equal weight is given to the two MSE loss terms during training.

The enhancement network is trained on the MSP longitudinal fingerprint dataset (rolled fingerprints only) [262], a subset of NIST SD 302 (rolled and plain fingerprints only) [78], and a dataset

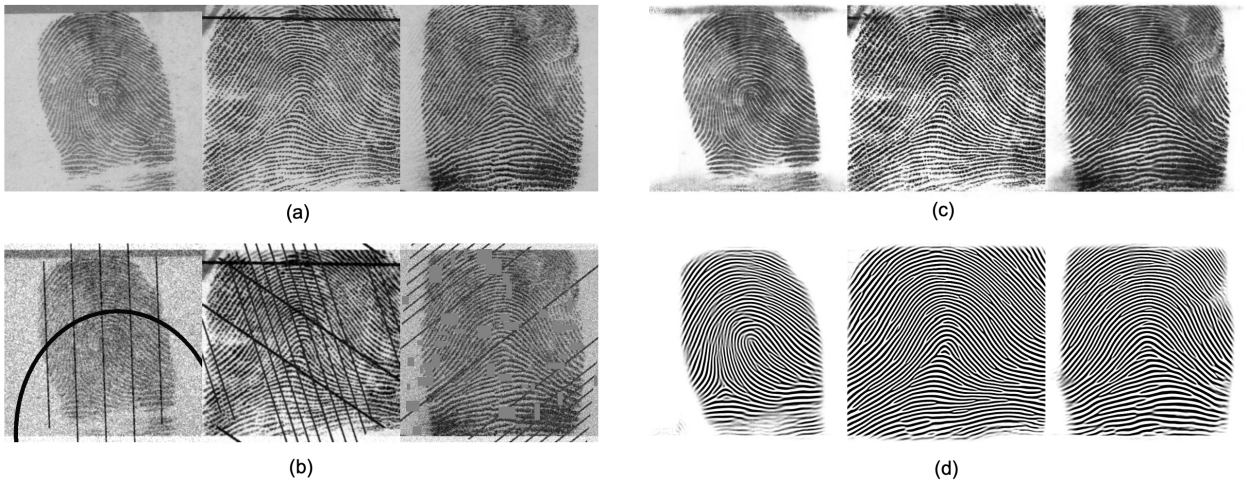


Figure 5.5 Example data augmentations to train the latent enhancement network. Random Gaussian blurring, Gaussian noise, downsampling, partial occlusions, and contrast adjustments are applied to rolled fingerprint images to generate low quality fingerprints that resemble characteristics of latent fingerprints. (a) original rolled fingerprints, (b) simulated latent fingerprints after data augmentations, (c) predicted enhanced output, and (d) predicted binary ridge image.

of plain fingerprint impressions referred to as the MSU Self-Collection. Details on number of fingers/images contained in each of these datasets are provided in Table 7.1. Ground truth binary images for all the training images are obtained using Verifinger v12.3. The network was trained on 2 Nvidia RTX A6000 GPUs for 11 epochs utilizing an initial learning rate of 0.001, polynomial learning rate schedule, and Adam optimizer. As is shown in section 5.5.2, despite not being trained on any real latent images, our enhancement network is able to outperform many of the existing latent enhancement methods in the literature. For illustration, example enhancements from each of these methods is shown in Figure 5.6.

5.3.2 Minutiae Extraction

Our minutiae extraction network consists of a ResNet50 backbone, self-attention transformer layers, and a series of transpose convolutional layers to predict a 12-channel minutiae map, a representation for minutiae points introduced in [31]. This minutiae map is converted to a list of (x, y, θ) locations for each minutiae point, and a set of 96×96 image patches centered around each minutiae are aligned based on the orientation θ and fed into a separate ResNet50 model to extract a set of descriptors associated with each minutiae. These descriptors are each 96-dimensional and

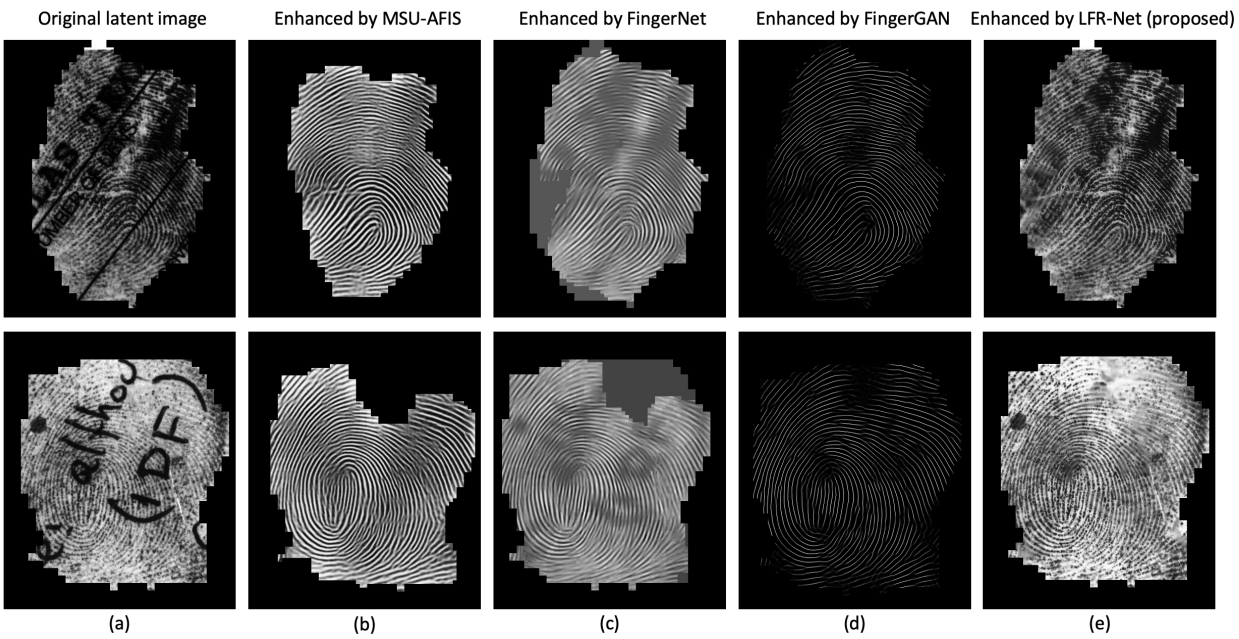


Figure 5.6 Two example comparisons of several baseline enhancement algorithms and the proposed enhancement network. (a) original latent images, (b) enhanced by MSU-AFIS [31], (c) enhanced by FingerNet [234], (d) enhanced by FingerGAN [271], and (e) enhanced by LFR-Net (proposed).

used in the minutiae similarity calculation when comparing two sets of minutiae points extracted from a given fingerprint image pair. Thus, in conjunction with the (x, y, θ) locations of each minutiae point and assuming m minutiae points in total, a given minutiae template M will be of dimension $M \in \mathcal{R}^{m \times 99}$. The architecture details of the minutiae extraction network are given in Table 5.2.

For matching minutiae points, we compute a similarity matrix between all Euclidean normalized minutiae descriptors and utilize the local similarity with relaxation (LSS-R) algorithm (as described in Minutiae Cylinder-Code (MCC) [34]) to refine and remove false correspondences. Finally, the cosine similarity between the descriptors of corresponding minutiae points are summed to yield a final minutiae similarity score. Due to the nature of latent fingerprint formation, we find it is extremely useful to align the minutiae points prior to extracting the minutiae descriptors. This step imparts the similarity calculation with rotation invariance, a critical factor in unconstrained latent fingerprint recognition.

Our minutiae extraction and descriptor networks are trained on the MSP (rolled fingerprints

Table 5.2 Architecture details for the minutiae extractor network. Batch normalization and ReLU activation are applied after each convolution, except for the last layer which uses a Sigmoid activation.

Layer Type	**Output Dim.**	**Parameters**
Conv2d	$64 \times 112 \times 112$	k=7x7, padding=3, stride=2
Conv2d	$256 \times 56 \times 56$	$\begin{bmatrix} \text{k=1} \times 1, \text{ch=64} \\ \text{k=3} \times 3, \text{ch=64} \\ \text{k=1} \times 1, \text{ch=256} \end{bmatrix}$ x3
Conv2d	$512 \times 28 \times 28$	$\begin{bmatrix} \text{k=1} \times 1, \text{ch=128} \\ \text{k=3} \times 3, \text{ch=128} \\ \text{k=1} \times 1, \text{ch=512} \end{bmatrix}$ x4
Conv2d	$1024 \times 14 \times 14$	$\begin{bmatrix} \text{k=1} \times 1, \text{ch=256} \\ \text{k=3} \times 3, \text{ch=256} \\ \text{k=1} \times 1, \text{ch=1024} \end{bmatrix}$ x6
MLP	384×196	in=1024, hid=1024, out=384
Self-Attention + MLP	384×196	in=384, hid=1536, out=384
Conv2d Transpose	$384 \times 28 \times 28$	k=2x2, stride=2
Conv2d	$384 \times 28 \times 28$	$\begin{bmatrix} \text{k=3} \times 3, \text{ch=384} \end{bmatrix}$ x2
Conv2d Transpose	$192 \times 56 \times 56$	k=2x2, stride=2
Conv2d	$192 \times 56 \times 56$	$\begin{bmatrix} \text{k=3} \times 3, \text{ch=192} \end{bmatrix}$ x2
Conv2d Transpose	$96 \times 112 \times 112$	k=2x2, stride=2
Conv2d	$96 \times 112 \times 112$	$\begin{bmatrix} \text{k=3} \times 3, \text{ch=96} \end{bmatrix}$ x2
Conv2d Transpose	$48 \times 224 \times 224$	k=2x2, stride=2
Conv2d	$48 \times 224 \times 224$	$\begin{bmatrix} \text{k=3} \times 3, \text{ch=48} \end{bmatrix}$ x2
Conv2d	$12 \times 224 \times 224$	k=1x1

only), NIST SD 302 (rolled and plain fingerprints only), and MSU Self-Collection (plain fingerprints only) training datasets. An MSE loss between predicted and ground truth minutiae points (obtained using the commercial Innovatrics v2.4.10 SDK) was used to supervise the minutiae extraction network. For training the minutiae descriptor model, minutiae patches of size 96×96 pixels were extracted from corresponding minutiae points between multiple impressions of each finger in the training set. To ensure reliability of ground truth corresponding minutiae patches, only corresponding minutiae points common among all impressions of the same finger were used and

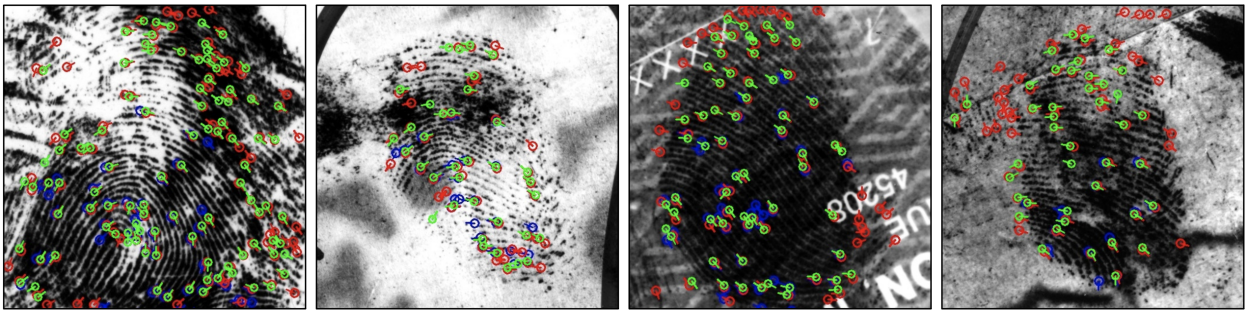


Figure 5.7 Visual comparison of minutiae extracted by our method (shown in green), Verifinger v12.3 (shown in red) and manually marked minutiae (shown in blue). Best viewed in color.

assigned a label for training. The Additive Angular Margin (ArcFace) loss function was used to supervise the descriptor model in classifying image patches belonging to the same minutiae point [56]. Both networks were trained on 4 Nvidia RTX A6000 GPUs for 56 epochs, with an initial learning rate of 0.0001, polynomial learning rate schedule, and Adam optimizer. A visual comparison of four example latent images annotated with minutiae from our minutiae extractor (shown in green), Verifinger v12.3 (shown in red), and manually marked minutiae (shown in blue) is provided in Figure 5.7. Due to the difficulty in manually marking latent minutiae points, usually very few minutiae are manually annotated. On the other hand, automatic minutiae extractors tend to detect many false (e.g., spurious) minutiae due to noise in the image. Nonetheless, compared to Verifinger, our method is detecting less spurious minutiae (as can be seen in the bottom two examples of Figure 5.7).

5.3.3 Virtual Minutiae Extraction

Due to the severely low quality of the ridges in many latent fingerprints, minutiae extraction is often unreliable and may produce many spurious minutiae and/or fail to extract any minutiae points at all. Therefore, in order to enforce local features within the image as part of matching, we utilize virtual minutiae, originally suggested in [31]. These virtual minutiae points are evenly spaced throughout the fingerprint area and use the estimated orientation field within the neighborhood of each point as the orientation assigned to each virtual minutiae point. Through an ablation study presented in section 5.5.2, we show the importance/utility of incorporating virtual minutiae into our pipeline.

For extracting virtual minutiae, we place a grid of virtual minutiae points at each (x,y) location of the segmented fingerprint area, separated by 16 pixels (in both x and y directions). The orientation of each 16x16 patch assigned to each virtual minutiae is estimated using the orientation field extraction algorithm described in [40]. Aligned image patches centered around each virtual minutiae are then fed to the same minutiae descriptor model described above to extract embeddings for each virtual minutiae. Since we are using the same minutiae descriptor extraction network, no additional training is required to obtain the virtual minutiae points. Assuming n virtual minutiae points are extracted in total, a given virtual minutiae template V will be of dimension $V \in \mathcal{R}^{n \times 99}$. The virtual minutiae similarity calculation between two virtual minutiae templates also utilizes the LSS-R matching algorithm [34].

5.3.4 Global Embedding Representation

For our global representation, we utilize the recently proposed AFR-Net architecture [97] which achieved high performance across a wide range of fingerprint types (rolled, plain, contactless) and sensors (optical, capacitive, etc.). AFR-Net is a combination of both CNN and ViT image recognition architectures, consisting of a shared CNN backbone and two separate classification heads (one CNN-based and the other utilizing attention blocks from ViT). The output of AFR-Net is two embeddings (Z_a and Z_c) of 384-dimensions each and the similarity score calculation is performed via a weighted sum of the normalized dot product between both embeddings of a fingerprint pair. For simplicity, we denote the AFR-Net embeddings as Z, a concatenation of the two individual embeddings (764-dimensional).

AFR-Net is trained on a diverse training set consisting of a combination of rolled fingerprints [71, 248,262], plain (i.e., slap) fingerprints [96], mixture of rolled and plain fingerprints [78], contactless (e.g., from mobile phone cameras) fingerprints [17, 53, 75], and synthetic latent fingerprints [251]. In total there are about 1.3 million images from 96,556 unique finger identities in training. Due to the lack of publicly available latent fingerprint datasets, we do not train on any real latent fingerprint databases and reserve the few latent dataset that we do have for evaluation. Interested readers are referred to [97] for complete training dataset details.

5.3.5 Minutiae Alignment of Global Embeddings

As proposed in [97], a strategy for improving the fingerprint representations obtained via deep learning networks is to align the regions of interest between two input images, remove background and other non-overlapping regions of the fingerprint areas in both images, and pass the aligned images back into the embedding network to yield new “refined" representations. In contrast to [97], where the local embeddings used to find corresponding regions of interest in both images are from an intermediate layer in the AFR-Net architecture, we directly use the minutiae correspondence between two images to compute the affine transformation which best aligns the image pair. In a sense, we are informing the global representation to focus on regions of the images which share many local similarities, in order to better distinguish between genuine pairs and close imposters. We also experimented with using virtual minutiae correspondences to compute the transformation but observed no significant change in the overall search accuracy to warrant the additional latency incurred by matching a much larger number of points.

5.3.6 Multi-Stage Search Strategy

Each of the feature sets in LFR-Net adds complimentary information for improving the reliability of a potential match, yet incurs an additional latency which can be prohibitively expensive on a large gallery size (e.g., N=100,000). Typically, computing the similarity between global, fixed length feature vectors (such as AFR-Net embeddings) is extremely fast compared to local feature matching (e.g., minutiae graph similarity computation); however, performance on small area latent fingerprints suffers without the use of local features. Therefore, we propose a multi-stage search paradigm which reduces the size of the returned candidate list before invoking expensive local feature matching (e.g., virtual minutiae similarity computation) to refine the final ranked candidate list.

Specifically, our hierarchical matching procedure consists of three stages. First, we return the top K (e.g., K=1,000) candidate matches using a fusion of AFR-Net similarity and minutiae matching. Next, we re-rank the top K candidates using virtual minutiae matching and obtain a smaller candidate list of size L (e.g., L=500). Finally, we align each probe image to each of its L

candidate gallery images (using an affine transformation computed from corresponding minutiae points) and obtain a new set of AFR-Net embeddings on the aligned images in order to further refine the final candidate list. An illustration of this multi-stage search strategy is shown in Figure 5.2. A discussion on the latency savings utilizing our three stage match procedure is given in section 5.4.5. The scores after each stage of matching are normalized to the range [0,1] based on a set of weights (w_1=0.4, w_2=0.4, w_3=0.18, and w_4=0.02) determined empirically on a validation set of latent fingerprints from the MSP latent database (which is separate from the MSP latent test dataset). The overall algorithm for LFR-Net is given in Algorithm 5.1.

5.4 Experimental Results

In this section, we report the performance of our latent fingerprint recognition pipeline across multiple latent fingerprint datasets, as well as other plain, rolled, and contactless fingerprint datasets to demonstrate the generalizability of our representations. First, we give the details of the datasets used in this study, followed by the closed-set and open-set identification results for several latent datasets, as well as the authentication performance across a diverse set of fingerprint sensors (e.g., capacitive, optical, etc.) and fingerprint type (plain, rolled, contactless, etc.). Next, we benchmark the performance of our enhancement network compared to previous enhancement methods, both in terms of minutiae detection accuracy and authentication performance of Verifinger v12.3 on each of the enhanced image outputs. Finally, we conclude with a discussion on the speed and computational efficiency of our recognition pipeline and the trade-offs in speed and accuracy given our multi-stage search strategy.

5.4.1 Datasets

Details for all training, validation, and test datasets used in this study are given in Table 7.1. Unlike previous latent fingerprint papers, we do not have access to a large private dataset of paired latent and rolled fingerprints (e.g., HiSign Latent Fingerprint database used in [102,258], consisting of 10,458 latent and mated rolled pairs). In fact, we do not use any latent fingerprint datasets for training, yet our system is able to achieve new SOTA accuracy on many latent test datasets. Since our system is not highly tuned for latent fingerprints, we are able to maintain SOTA accuracy on

Algorithm 5.1 Return a ranked candidate list, given an input latent fingerprint probe (I_p) and gallery of rolled fingerprint images (I_G) using the proposed LFR-Net matcher.

procedure MATCH(I_p, I_G)
  // Initialize score weights
  $w_1, w_2, w_3, w_4 := 0.4, 0.4, 0.18, 0.02$

  // Initialize no. candidates passed to 2^{nd} stage.
  $K := 1000$

  // Initialize no. candidates passed to 3^{rd} stage.
  $L := 500$

  // No. candidates in gallery.
  $N \leftarrow len(I_G)$

  // Initialize score lists.
  $S_1, S_2, S_3 := [0] * N, [0] * K, [0] * L$

  // Extract gallery and probe features.
  $M_p, V_p, Z_p \leftarrow Extract(I_p)$
  $M_G, V_G, Z_G \leftarrow Extract(I_G)$

  // Stage 1 Matching
  for i in range(N) **do**
    $M_g, Z_g := M_G[i], Z_G[i]$
    $S_1[i] \leftarrow w_1 m_{simi}(M_p, M_g) + w_2 \frac{(Z_p^T \cdot Z_g)}{|Z_p||Z_g|}$
  $I_G^1, M_G^1, V_G^1, Z_G^1 \leftarrow SortAndFilterCandidates(S_1)$

  // Stage 2 Matching
  for i in range(K) **do**
    $M_g, V_g, Z_g := M_G^1[i], V_G^1[i], Z_G^1[i]$
    $S_2[i] \leftarrow w_1 m_{simi}(M_p, M_g) + w_2 \frac{(Z_p^T \cdot Z_g)}{|Z_p||Z_g|}$
    $+w_3 m_{simi}(V_p, V_g)$
  $I_G^2, M_G^2, V_G^2, Z_G^2 \leftarrow SortAndFilterCandidates(S_2)$

  // Stage 3 Matching
  for i in range(L) **do**
    $M_g, I_g, Z_g, V_g := M_G^2[i], I_G^2[i], Z_G^2[i], V_G^2[i]$
    $Z_p', Z_g' \leftarrow Realign(I_p, M_p, I_g, M_g)$
    $S_3[i] \leftarrow w_1 m_{simi}(M_p, M_g) + w_2 \frac{(Z_p^T \cdot Z_g)}{|Z_p||Z_g|}$
    $+w_3 m_{simi}(V_p, V_g) + w_4 \frac{(Z_p'^T \cdot Z_g')}{|Z_p'||Z_g'|}$
  $I_G^3, M_G^3, V_G^3, Z_G^3 \leftarrow SortAndFilterCandidates(S_3)$

  // Return sorted candidate list.
  return I_G^3

Table 5.3 Fingerprint datasets used in this study.

Train Datasets	**# Fingers**	**# images**
MSP Rolled† [262]	37,411	447,988
NIST SD 302 (N2N)‡ [78] (plain and rolled prints)	1,600	20,008
MSU Self-Collection (plain prints)	4,582	57,813
Validation Datasets	**# Fingers**	**# Images**
NIST SD 302 (N2N)‡ [78] (plain and rolled prints)	200	2,528
MSP Latent† [262]	933	2,030
Test Datasets	**# Fingers**	**# Images**
NIST SD 14 [246]	2700	5,400
NIST SD 302 (N2N)‡ [78]	200	2,548
NIST SD 302 Latents (N2N Latents) [78]	1,019	3,793
IIIT-D MOLF [209]	1,000	12,400
MSP Latent [262]	933	1,866
NIST SD 27 [83]	258	516
PolyU Contactless 2D to Contact-based 2D Database [145]	160	960
ZJU Finger Photo and Touch-based Database [93]	824	19,776

† The MSP Rolled and MSP Latent datasets are completely disjoint and distinct in terms of finger identities.
‡ The train, validation, and test splits of N2N are disjoint and distinct in terms of finger identities.

Table 5.4 Closed-set identification (1:N comparison) results of three matchers, including the proposed LFR-Net.

Model	**Feature Extraction latency‡**	**Match latency†**	**Template size (latent/rolled)**	**Closed-Set Rank 1 Retrieval Rate (%)**				
				NIST SD 27	**N2N Latent**	**MSP Latent**	**MOLF DB1/DB4**	**MOLF DB2/DB4**
MSU-AFIS [31]	2,586ms	0.093ms	308KB / 56KB	61.63	29.78	67.64	44.84	27.86
AFR-Net [97]	6.86ms	0.002ms	3KB / 3KB	39.92	21.95	59.27	42.41	38.48
LFR-Net (proposed)	553ms	0.068ms*	307KB / 401KB	**84.11**	**54.36**	**84.35**	**70.43**	**62.86**

‡ Computed on an Nvidia RTX A6000 GPU.
† 128 threads on an AMD EPYC 7543 32-Core Processor.
∗ Time computed using 3-stage matching with K=1,000 and L=500. Recall, K is the no. of candidates from the gallery passed from stage 1 to stage 2 and L is the no. of candidates passed from stage 2 to stage 3.

Table 5.5 Rank-1 Retrieval Rate (%) of the proposed LFR-Net on NIST SD 27 gallery sizes ranging from 50,000 to 250,000.

Gallery Size (N)	**50K**	**100K**	**150K**	**200K**	**250K**
Rank-1 Retrieval Rate (%)	85.66	84.11	83.33	82.56	82.17

rolled, plain, and contactless fingerprints as well.

5.4.2 Identification Results

Particularly relevant to the use case of latent fingerprint recognition is the ability to quickly and accurately search a large database of known fingerprints given an input probe latent fingerprint

Table 5.6 Open-set comparison of FNIR at FPIR=0.02 across 5 latent dataset evaluations (lower is better).

Method	NIST SD 27	N2N Latent	MSP Latent	MOLF DB1/DB4	MOLF DB2/DB4
MSU-AFIS [31]	0.72	0.86	0.69	0.76	0.87
LFR-Net (proposed)	**0.50**	**0.74**	**0.44**	**0.60**	**0.68**

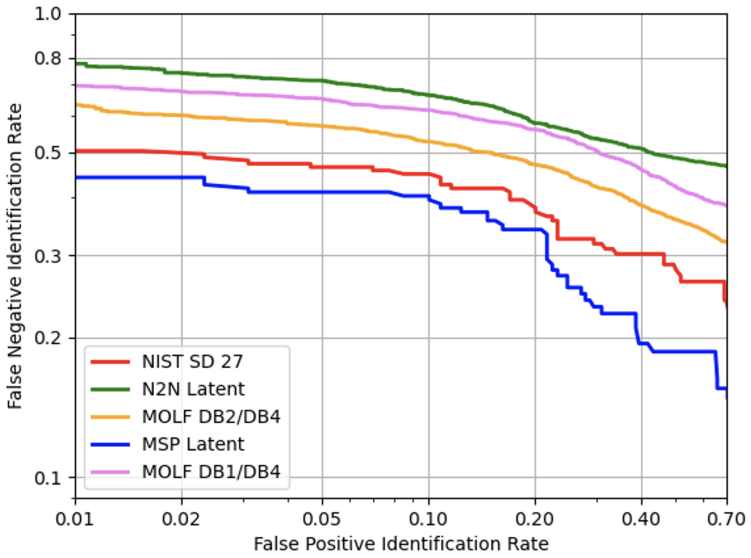


Figure 5.8 Open-set performance of LFR-Net with a gallery of 100K. Only 50% of the latent probes in each dataset have mates in the gallery.

image. To benchmark the performance of our proposed pipeline for latent to rolled fingerprint recognition, we compute the closed-set and open-set identification results across four different latent fingerprint datasets against a gallery of 100K rolled fingerprints. Each fingerprint image in the gallery is from a unique finger and is separate from the rolled mates included in each latent database. Furthermore, all gallery fingerprints used in these experiments are from a separate set of the MSP forensic database [262] and were not used in training.

A comparison of the rank-1 retrieval rate of the proposed method and two baseline algorithms for latent fingerprint recognition is shown in Table 5.4. The two baseline methods include i.) the original AFR-Net as proposed in [97] and ii.) an optimized version of the MSU-AFIS latent recognition pipeline proposed in [31]. Currently, none of the commercial latent fingerprint vendors provide their latent fingerprint SDK to include in the evaluation. It is reported that Verifinger SDK

v13 may have a latent matcher, but it is not yet available to developers at this time. Additionally, many other previous latent fingerprint algorithms in the literature either have not made the source code publicly available or take prohibitively long to run the evaluation on our large size of gallery. Where available, we have included the performance of these baseline methods using the numbers reported in those publications in Table 5.1. Not surprisingly, AFR-Net underperforms across each of the latent datasets compared to both MSU-AFIS and our proposed pipeline. MSU-AFIS performs reasonably well across each dataset because of its use of minutiae and virtual minutiae which, according to our ablation table in section 5.5.2, makes a significant difference in the accuracy for latent to rolled comparisons. Nonetheless, our method, LFR-Net, outperforms all baseline methods due to a combination of improved enhancement, segmentation, and fusion of both local and global embeddings. In particular, our average rank-1 retrieval rate across the four datasets is 71.22%, compared to the average rank-1 performance of MSU-AFIS of 46.19%. Out of the published results on NIST SD 27, LFR-Net outperforms the next best method of Gu et al. 84.11% to 70.1% at rank-1.

For time and space constraints, most of our experiments are conducted on a background gallery of 100,000 unique rolled fingerprints; however, in order to investigate the scalability of LFR-Net to larger gallery sizes, we conducted closed-set identification experiments using NIST SD 27 on gallery sizes ranging from N=50,000 to N=250,000 rolled distractors. The results given in Table 5.5 suggest that the decline in rank-1 retrieval rate up to a gallery size of 250,000 starts to converge toward 82%, down from an initial 85.66% for gallery size of 50,000. As part of future work, we will investigate this trend up toward a gallery size of 1,000,000 unique fingers, which is more indicative of practical real-world applications.

To the best of our knowledge, previous studies on latent to rolled fingerprint matching have only reported closed-set identification results. With our improved performance across many of the latent to rolled datasets, we also report open-set identification results where 50% of the probes from each of our datasets are randomly selected to have no corresponding mates in the gallery. A plot

https://neurotechnology.com/verifinger.html

Table 5.7 Authentication (1:1 comparison) results of three matchers, including the proposed LFR-Net.

Model	TAR (%) @ FAR=0.01%					EER (%)				
	NIST SD 14	NIST SD 302	PolyU‡	ZJU‡	NIST SD 27†	NIST SD 14	NIST SD 302	PolyU‡	ZJU‡	NIST SD 27†
Verifinger v12.3	99.93	93.26	95.39	96.88	55.04	**0.04**	2.52	1.01	0.83	12.91
AFR-Net [97]	99.93	95.42	98.04	98.78	59.69	**0.04**	2.03	**0.34**	0.50	11.52
LFR-Net (stage 1)	**99.96**	**95.56**	**98.61**	**99.00**	**68.99**	**0.04**	1.96	0.45	**0.45**	7.74
LFR-Net (stage 1&2)	99.93	94.30	**98.61**	**99.00**	**68.99**	**0.04**	**1.87**	0.45	**0.45**	**6.58**

† Using LFR-Net preprocessing and segmentation.

‡ Contactless images are preprocessed using the enhancement and unwarping method proposed in [93].

of false negative identification rate (FNIR) vs. false positive identification rate (FPIR), computed for rank-1 retrieval, for LFR-Net across all five latent evaluations is given in Figure 5.8, where a comparison of FNIR @ FPIR=0.02 with MSU-AFIS is given in Table 5.6.

5.4.3 Authentication Performance

Due to the challenges inherent to latent fingerprint recognition, all the existing systems in the open literature are highly tuned to perform well for latent to rolled comparison and/or require expensive feature extraction and matching times due to the additional features required to achieve high accuracy. However, our system stands out in that the representations learned (both local and global embeddings) are generalizable across a wide range of fingerprint image characteristics, as demonstrated in the authentication results shown in Table 5.7. LFR-Net is competitive with and even outperforms the commercial fingerprint SDK Verifinger v12.3 on several datasets.

Furthermore, our algorithm can be tuned to vary the latency for both feature extraction and matching depending on the confidence required and/or difficulty of the fingerprint image domain of interest. For example, both the feature extraction and matching latencies could be significantly decreased for rolled-to-rolled or plain-to-plain fingerprint matching by utilizing just the AFR-Net embeddings, which achieves very competitive authentication performance across all of the datasets. Similarly, at a modest cost in latency, one could also incorporate minutiae features for a slight boost in accuracy, as is done in stage 1 of LFR-Net. Virtual minutiae are effective in improving latent

to rolled matching accuracy; however, they introduce additional latency which is not required to achieve high accuracy across rolled and plain fingerprint datasets.

5.4.4 Latent Enhancement Performance

For a comparison of our enhancement method with several previous SOTA latent enhancement methods (FingerGAN [271], FingerNet [234], and MSU-AFIS [31]) we computed the statistics of false (spurious) and correctly predicted minutiae using the manually marked ground truth provided for NIST SD 27 from [77]. Specifically, we individually enhanced all 258 latent images from NIST SD 27 using each of the enhancement methods, extracted the minutiae points of the enhanced images using Verifinger v12.3, and compared the extracted minutiae to the human annotated ground truth minutiae points. We consider a correctly detected minutiae as one in which the type is the same, (x,y) location is within 10 pixels, and the angle difference is less than 10 degrees compared to a ground truth minutiae. These thresholds are motivated from a previous study on the robustness of minutiae-based matchers, which showed that the performance of minutiae matching starts to decline with minutiae perturbations outside these ranges [94]. Results in Table 5.8 show that our enhancement network outperforms the previous methods in terms of number of correctly predicted minutiae, while also introducing fewer spurious minutiae than the next best method, FingerGAN. Interestingly, the number of spurious predicted minutiae is the lowest for the unenhanced, original latent images. The increase in spurious minutiae may, at least partially, be attributed to each enhancement method hallucinating fingerprint ridges where there are none; however, the number of spurious minutiae increasing dramatically across all methods suggests that the increase could be attributed in some part to missed minutiae during the manual markup process due to the difficulty in annotating minutiae for unenhanced latent fingerprints. Nonetheless, according to previous research, spurious minutiae have much less of an effect on overall matching accuracy as does failing to detect correct minutiae [94]. In fact, we can see this is indeed the case in the improved authentication (1:1 matching) performance of Verifinger on our enhanced NIST SD 27 images (TAR=65.12% at FAR=0.1%) compared to the original images (TAR=57.75% at FAR=0.1%), where a total of 258 genuine scores and 66,564 imposter scores were computed.

Table 5.8 Number of correctly predicted minutiae and spurious minutiae introduced for the 258 latent prints in NIST SD 27 before and after enhancement by several methods. Predicted minutiae were extracted using Verifinger v12.3, and ground truth minutiae were manually marked by [77].

Method	No. Correctly Predicted	No. Spurious Predictions	TAR (%) @ FAR=0.01% (0.1%)	Rank-1 (%) with gallery of 100K
Original Images	2,606	**3,676**	51.94 (57.75)	56.59
Enhanced by MSU-AFIS [31]	2,329	4,678	38.76 (44.96)	48.06
Enhanced by FingerNet [234]	2,460	5,147	39.15 (47.29)	47.29
Enhanced by FingerGAN [271]	2,939	7,385	52.71 (57.36)	**58.14**
Enhanced by LFR-Net (Proposed)	**3,118**	5,536	**55.04 (65.12)**	**58.14**

Interestingly, the rank-1 performance using Verifinger is the same for our enhanced images as those of FingerGAN [271]; however, we have the additional advantage that our enhanced images are not introducing a domain shift with respect to the original, gray-scale fingerprint images. This means that other components of our pipeline (e.g., minutiae descriptor and AFR-Net) can also benefit from the enhancement without the need to be re-trained. For example, the rank-1 performance of AFR-Net on NIST SD 27 enhanced by LFR-Net improves from 39.92% to 52.33% without re-training the network, but the performance drops to 6.20% using FingerGAN enhanced images.

5.4.5 Computational Efficiency

Latency is a crucial aspect for large-scale identification applications, which tends to be in competition with accuracy. Thus, we were motivated to find a balance between accuracy and speed using a multi-stage search protocol, which has also been explored in previous works on fingerprint identification [68]. For a quantitative analysis on the latency of our approach, we will denote the size of our gallery as N (e.g., N=100,000) and the size of our probe dataset as Q (e.g., Q=258 in the case of NIST SD 27). Furthermore, since our algorithm consists of three stages of matching with variable number of top candidates per probe passed to subsequent stages, we will denote the number of candidates per probe image passed from the first stage to the second stage as K and the number of candidates passed to the third stage as L.

For our first stage matching, we utilize only AFR-Net and minutiae features to obtain a short list of top K candidates from the gallery for each probe fingerprint image. This stage takes on average t_1=0.015 ms for a single latent to rolled comparison when utilizing 128 threads on an AMD EPYC 7543 32-Core Processor, where a total of NxQ comparisons are computed. In the second stage, we utilize virtual minutiae scores to re-rank the K list of candidates per latent and return a further condensed list of top L candidates to pass to the third stage. Here, a single virtual minutiae comparison between a latent and rolled image pair takes on average t_2=0.984 ms, where a total of KxQ comparisons are computed. Finally, our third stage consists of re-aligning each of the L candidate images for each probe using the pairwise minutiae correspondences and recomputing AFR-Net scores for each pair. In this stage, there are a total of LxQ comparisons required, where each realignment plus AFR-Net inference per comparison takes an average of t_3=8.626 ms. Note, the latency of stage 1 and stage 2 depends on the number of minutiae and virtual minutiae extracted per latent probe, respectively. The latency numbers reported here are computed for NIST SD 27 against a gallery augmented by 100,000 rolled fingerprints, where the average number of minutiae and virtual minutiae extracted per latent image is 45 and 363, respectively, and the average number of minutiae and virtual minutiae per rolled fingerprint is 119 and 886, respectively. In total, the average latency t per comparison for the entire three stage matching process can be computed using equation 5.1:

$$t = t_1 + \frac{K}{N}t_2 + \frac{L}{N}t_3 \tag{5.1}$$

Using equation 5.1 with N=100,000, K=1000 and L=500, the average latent to rolled comparison across each of the four latent datasets for our full matching pipeline takes about t=0.068 ms. As mentioned previously, the filtering of our candidate lists in each stage does incur some accuracy trade-off; however, we find that filtering 99% of the candidate list prior to stage 2 (with K=1,000 and N=100,000) leads to no difference in rank-1 retrieval rate for NIST SD 27 and only about a 1% decrease in accuracy at higher ranks. A plot of the Cumulative Match Characteristic (CMC)

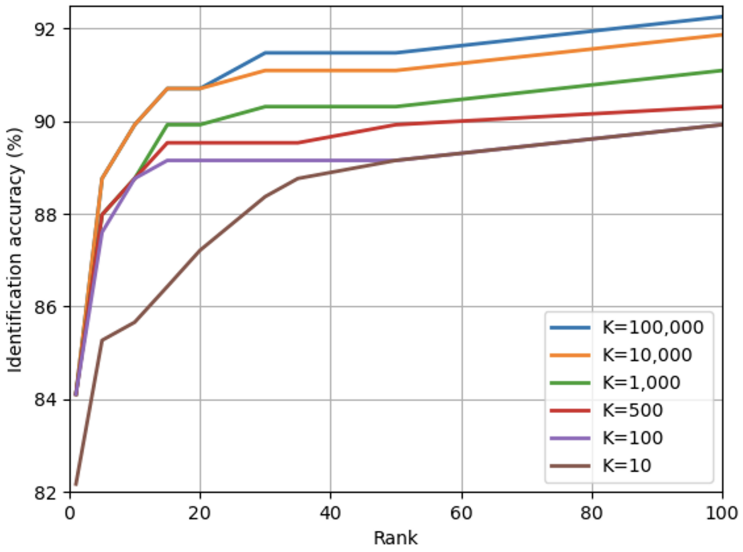


Figure 5.9 Search results on NIST SD 27 (100K gallery) as the no. of candidates (K) sent to stage 2 is varied. Reducing K to 1% of the gallery results in a speed up of 0.352 ms to 0.068 ms (5.2× faster) per comparison with no change in rank-1 accuracy and only ~ 1% decrease at higher ranks.

for NIST SD 27 on a gallery of 100,000 as the value of K is varied from 100,000 to 10 is shown in Figure 5.9.

The feature extraction speed is often less of a concern for fingerprint recognition since templates for the gallery can be extracted offline prior to matching; however, is still important in cases of updating the gallery for future improvements to the system. Nonetheless, our method is significantly faster compared to the baseline MSU-AFIS algorithm, taking just 553 ms on average per latent image or 1.88 images per second. In terms of template size, our algorithm is comparable to MSU-AFIS for latents; however, for rolled templates, MSU-AFIS performs several template compression and quantization techniques to reduce the size of the templates compared to ours, which can also be incorporated into our algorithm in future work.

5.5 Discussion

In this section we discuss some of the failure cases of our pipeline and present plausible strategies to improve on these challenging cases. Furthermore, we present an ablation analysis to evaluate the contribution of each component of our pipeline to the overall identification accuracy.

5.5.1 Failure Case Analysis

Despite the SOTA performance of our algorithm across all five latent evaluation fingerprint datasets utilized, there are still cases in which our algorithm fails to return the correct mate at rank one (see Figure 5.10 (c) and (d)). Figure 5.10 (a) and (b) show examples where the system was successful in returning the correct mate at rank-1, demonstrating the benefit of the enhancement network in removing some occlusions and enhancing the inter-ridge separation. However, as examples (c) and (d) show, there are many challenges that need to be tackled. One cause of failure, demonstrated in (c), is poor segmentation, where a large portion of the fingerprint image is cut-off. Additional failures can be attributed to noisy background and overlapping latent patterns, very low ridge-valley contrast, and extreme rotations. In case of small area latent fingerprints, it becomes difficult to estimate the correct rotation to align latent fingerprints prior to global feature extraction. A simple way to overcome this difficulty could be to rotate the latent image 4 times by 90° increments and take a max fusion of the global similarity scores obtained when matching each of the four rotated images with all the images in the gallery. Perhaps an even better approach would be to make the global embeddings inherently rotation invariant, which will be one focus of our future work.

5.5.2 Ablation Study

To justify the use of each component (enhancement, minutiae, virtual minutiae, global embedding, and realignment stage), we perform an ablation analysis as each additional module is added to the matching pipeline. The results are given in Table 5.9, which show improved search performance with the addition of each module. We observe a significant jump in accuracy with the incorporation of local minutiae features, another significant jump in accuracy with virtual minutiae, and a final improvement in using the realignment stage. The performance across all datasets starts to saturate after the second stage matching with virtual minutiae, where the third stage (realignment) adds the most noticeable benefit for datasets with extreme rotations (e.g., N2N Latent). We also see significant improvements in rank-1 retrieval with using our enhancement network vs. without any enhancement. For example, the performance on NIST SD 27 improves from 72.87% without

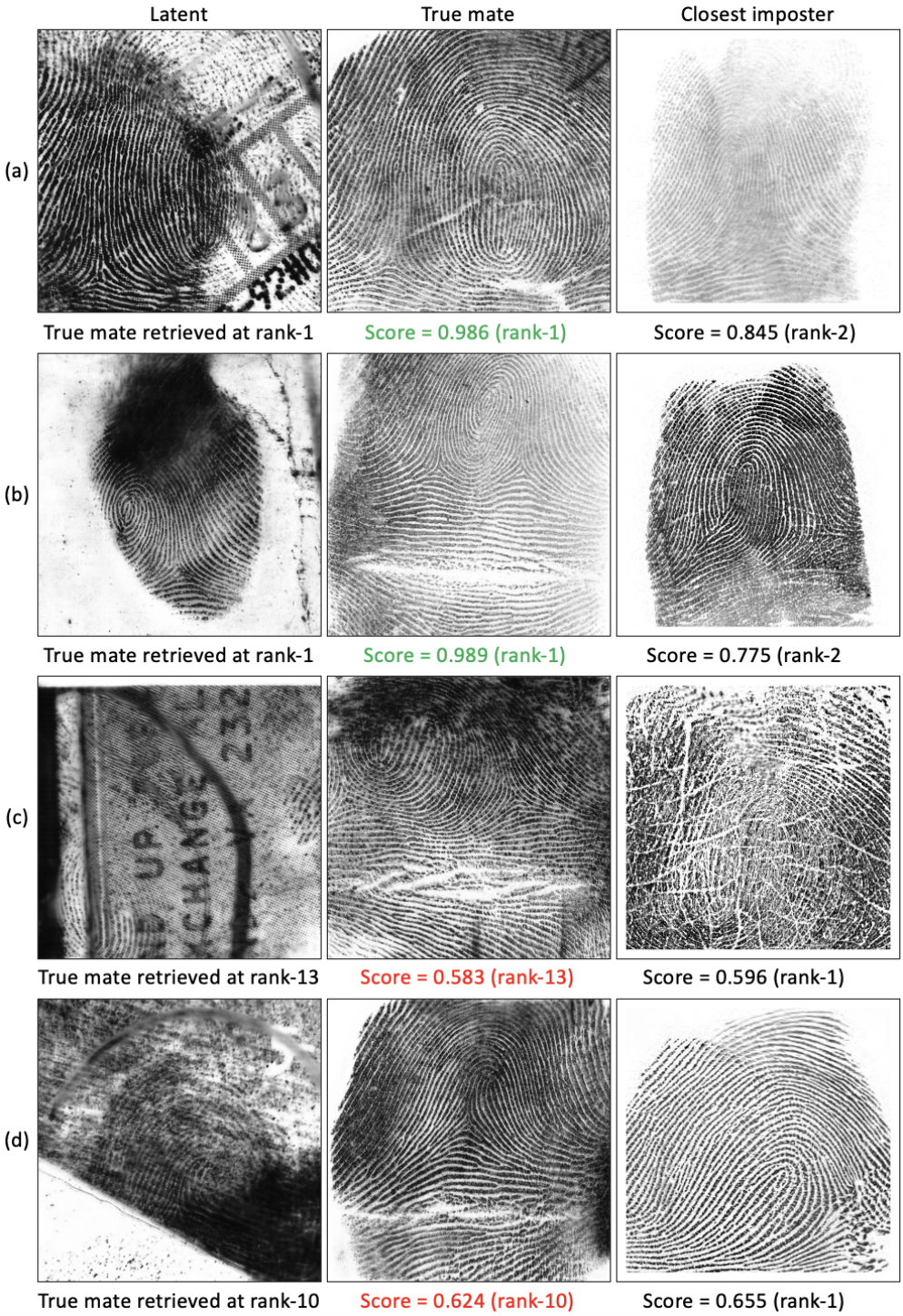


Figure 5.10 Example success (a and b) and failure (c and d) cases of the proposed LFR-Net on the NIST SD 27 latent database.

Table 5.9 LFR-Net ablation study.

Modules					Rank-1 accuracy (%) on a gallery of 100K				
Global Emb.	**Minu.**	**Virtual Minu.**	**Realign**	**Enhance**	**NIST SD 27**	**N2N Latent**	**MSP Latent**	**MOLF (DB1/DB4)**	**MOLF (DB2/DB4)**
✓					39.92	21.95	59.27	42.41	38.48
	✓				57.75	46.47	72.45	45.05	33.36
✓	✓				65.12	46.58	78.67	54.86	45.20
✓	✓	✓			72.48	49.21	80.71	60.34	50.25
✓	✓	✓	✓		72.87	50.25	81.78	60.77	50.52
✓				✓	52.33	23.00	60.34	53.48	49.85
	✓			✓	67.69	51.66	78.24	60.52	50.32
✓	✓			✓	75.58	51.12	81.46	67.18	59.68
✓	✓	✓		✓	**84.11**	53.50	83.92	70.14	62.68
✓	✓	✓	✓	✓	**84.11**	**54.36**	**84.35**	**70.43**	**62.86**

enhancement to 84.11% with enhancement.

5.6 Conclusion

In this chapter, we presented a pipeline for end-to-end latent fingerprint recognition and demonstrated its SOTA performance across five different latent fingerprint evaluations (for both closed-set and open-set identification), as well as its generalization across several rolled, plain, and contact to contactless fingerprint datasets. Our network incorporates a novel use of both local (minutiae and virtual minutiae) and global (AFR-Net) embeddings for improved latent fingerprint recognition. We also present a multi-stage search strategy to decrease the time required for large-scale identification, which is adaptable for a desired trade-off in accuracy and search speed. Despite the performance improvement achieved by the methods proposed in this chapter, there still exists a gap in performance compared to controlled rolled to rolled fingerprint recognition. One of the significant factors limiting the performance of latent fingerprint recognition is the lack of large-scale, publicly available latent datasets for training. In fact, a lack of publicly available fingerprint recognition data is a significant inhibitor to progress in many aspects of fingerprint recognition. In the next chapter, we turn to synthetic fingerprint generation to help alleviate this challenge in one of these notable applications, fingerprint spoof (i.e., presentation attack) detection, where limited data is of particular consequence.

5.7 Acknowledgment

Parts of this research were supported by a grant from the Department of Homeland Security via The Criminal Investigations and Network Analysis Center (CINA) at George Mason University.

CHAPTER 6

SYNTHETIC FINGERPRINT SPOOF IMAGES

A major limitation to advances in fingerprint presentation attack detection (PAD) is the lack of publicly available, large-scale datasets - a problem which has been compounded by increased concerns surrounding privacy and security of biometric data. Furthermore, most state-of-the-art PAD algorithms rely on deep networks which perform best in the presence of a large amount of training data. This chapter aims to demonstrate the utility of synthetic (both bona fide and PA style) fingerprints in supplying these algorithms with sufficient data to improve the performance of fingerprint PAD algorithms beyond the capabilities when training on a limited amount of publicly available "real" datasets. First, we provide details of our approach in modifying a state-of-the-art generative architecture to synthesize high quality bona fide and PA fingerprints. Then, we provide quantitative and qualitative analysis to verify the quality of our synthetic fingerprints in mimicking the distribution of real data samples. We showcase the utility of our synthetic bona fide and PA fingerprints in training a deep network for fingerprint PAD, which dramatically boosts the performance across three different evaluation datasets compared to an identical model trained on real data alone. Finally, we demonstrate that only 25% of the original (real) dataset is required to obtain similar detection performance when augmenting the training dataset with synthetic data. We make our synthetic dataset and model publicly available to encourage further research on this topic: https://github.com/groszste/SpoofGAN.

6.1 Introduction

Fingerprint recognition has had a long history in person identification due to the purported uniqueness and permanence of fingerprints, originally pointed out by Sir Francis Galton in his 1892 book titled *Finger Prints* [81] and reaffirmed in many works over the last century, including the well-known studies on the individuality and longitudinal permanence of fingerprint recognition [190, 262]. Clearly, a significant contributor to their widespread adoption is the high level of

This chapter was previously published as S. A. Grosz and A. K. Jain, "SpoofGAN: Synthetic Fingerprint Spoof Images", IEEE Transactions on Information Forensics and Security, vol. 18, pp. 730-743, 2023.

Figure 6.1 Example fabricated fingerprint PAs of various materials and corresponding fingerprint impressions captured on a CrossMatch Guardian200 fingerprint reader. (a) PlayDoh PA, (b) printed paper PA, (c) latex PA, (d) fingerprint impression from the PlayDoh artifact, (e) fingerprint impression from the printed paper artifact, and (f) fingerprint impression from the latex artifact.

verification performance achieved by state-of-the-art (SOTA) algorithms for automated fingerprint recognition. However, despite the impressive accuracy achieved to date by the top-performing fingerprint recognition algorithms, there remain many ongoing efforts to further improve the capabilities of fingerprint recognition systems - especially in terms of recognition speed and system security. As a result, there has been a recent push toward deep neural network (DNN) based models for fingerprint recognition [26, 68, 93, 139, 143, 217, 219]. These compact, fixed-length embeddings can be matched efficiently and combined with homomorphic encryption for added security [74]. For a more exhaustive account of existing deep learning approaches to fingerprint recognition and other biometric modalities, interested readers are encouraged to consult one of the many surveys on deep learning in biometrics (e.g., [172, 226]).

Indeed, this push toward DNN-based fingerprint recognition comes in the wake of the success demonstrated in the face recognition domain in applying DNN models to face recognition, which was aided by the availability of large-scale face recognition databases which were easily crawled from the web despite the many ethical and privacy concerns which have led to many of these datasets to be recalled today. Arguably, at least in part, the reason for the delayed adoption of DNNs for fingerprint recognition has been the lack of publicly available, large-scale fingerprint recognition datasets and increased scrutiny over privacy of biometric data, which has led to many works to

Today's top-performing algorithm in the FVCongoing 1:1 hard benchmark achieved a False Non-Match Rate (FNMR) of just 0.626% at a False Match Rate (FMR) of 0.01% [63]

generate synthetic fingerprint images [6, 9, 11, 19, 21, 23, 36, 71, 76, 125, 171, 173, 198, 252, 268].

Similarly, there has been an increased interest in DNN-based models for fingerprint PAD (i.e., spoof detection), where the scale and amount of publicly available data is also limited. Table 6.1 gives a list of the publicly available fingerprint PA datasets. Compared to the largest, public fingerprint recognition dataset, NIST Special Database 302 [78], which contains fingerprints from 2,000 unique fingers, the largest publicly available fingerprint PA datasets, e.g., the LivDet competition datasets, contain at most 1,000 unique fingers (for the Swipe sensor in LivDet 2013 [86]). Compounding the problem is the difficulty in collecting large-scale fingerprint PA datasets due to the increased time and complexity in fabricating and imaging artifacts mimicking realistic fingerprint ridge-valley structures, motivating the potential of synthetic data as a viable alternative. However, to the best of our knowledge, there does not exist a synthetic fingerprint PA generator to fill the gap between the amount of publicly available fingerprint PA data and training of data-hungry deep learning-based models.

To address the lack of large-scale fingerprint PA datasets, we propose SpoofGAN. Inspired by the impressive results of the recently proposed PrintsGAN [71], SpoofGAN is a multi-stage generative architecture to fingerprint generation. SpoofGAN is different from PrintsGAN in the following ways:

- Generation of plain print fingerprints, which compared to the rolled prints generated by PrintsGAN, are more representative of the publicly available PA fingerprint datasets and exhibit different textural characteristics, distortions, etc.
- Ability to synthesize representations of both bona fide (i.e., live) and PA fingerprint images of the same finger.
- Replacing the learned warping and cropping module with a statistical, controllable non-linear deformation model to synthesize multiple, realistic impressions per finger. This allows us to control the degree of distortion applied.

Swipe sensors are no longer in vogue after Apple introduced the “area capacitive sensor" in Touch ID.

In this work, we make the distinction that real fingerprint images are those captured on a fingerprint reader by either a real human finger or physical artifact (presentation attack) which mimics a fingerprint ridge structure, whereas synthetic fingerprint images are digital renderings of fingerprint images. However, there can be both real bona fide and real PA fingerprint images as well as synthetic representations of bona fide and synthetic representations of PA fingerprint images. For clarification, we use the following four classifications:

- *Real bona fide* fingerprint image: fingerprint images captured from a real human finger on a fingerprint reader.
- *Real PA* fingerprint image: fingerprint images captured from a presentation attack artifact on a fingerprint reader.
- *Synthetic bona fide* fingerprint image: synthetic images that mimic the distribution of fingerprint images captured from a real human finger.
- *Synthetic PA* fingerprint image: synthetic images that mimic the distribution of fingerprint images captured from presentation attack artifacts.

We validate the realism of our synthetic bona fide and PA images through extensive qualitative and quantitative metrics including NFIQ2 [228], minutiae statistics, match scores from a SOTA fingerprint matcher, and T-SNE feature space analysis showing the similarity of real bona fide and PA embeddings to the embeddings of our synthetic bona fide and PA fingerprints. Besides verifying the realism of our synthetic PA generator, we also show how SpoofGAN fingerprints can be used to train a DNN for fingerprint PAD. We show this by improving the performance of a PAD model by augmenting an existing fingerprint PA dataset with additional samples from our synthetic generator. We also open the door to jointly optimizing for fingerprint PAD and recognition in an end-to-end learning framework with our ability to generate a large-scale dataset of multiple impressions per finger of both bona fide and PA examples.

More concisely, the contributions of this research are as follows:

- A highly realistic plain print synthetic fingerprint generator capable of generating multiple impressions per finger.

- The first, to the best of our knowledge, synthetic fingerprint PA generator which is capable of producing synthetic representations of both bona fide and PA impressions of the same finger. This opens the door to joint optimization of fingerprint PAD and recognition algorithms.
- Quantitative and qualitative analysis to verify the quality of our generated bona fide and PA fingerprints.
- Experiments showcasing improved fingerprint PAD on both seen and unseen PA material types when augmenting existing fingerprint datasets with our synthetic bona fide and PA fingerprints.
- We release our code and a database of SpoofGAN images to encourage further research in this area https://github.com/groszste/SpoofGAN.

6.2 Related Work

6.2.1 Fingerprint Presentation Attack Detection

One significant risk to the security of fingerprint recognition systems is that of presentation attacks, defined by the international standard ISO/IEC 30107-1:2016 as a "presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system" [117]. The most common type of presentation attacks are spoof attacks, i.e., physical representations of finger-like structures aimed at either mimicking the fingerprint ridge-valley structure of another individual or subverting the user's own identity. Spoof attacks may come in many different forms and materials such as those shown in Figure 6.1.

Several hardware-based and software-based solutions to detecting spoof attacks have been proposed. Hardware-based solutions include specialized sensors that leverage various "liveness" cues at the time of acquisition, such as conductivity of the material/finger, sub-dermal imaging, and multi-spectral lighting [12, 43, 67, 135, 221, 236]. On the other hand, software-based solutions typically rely on only the information captured in the grayscale image acquired by the fingerprint reader [41, 84, 85, 164, 167, 177, 188]. Despite the limited publicly available fingerprint PA data, many of the state-of-the-art software-based solutions to fingerprint PAD leverage convolutional

In this work, we use the terms spoof and presentation attack interchangeably.

Table 6.1 Publicly available fingerprint PA datasets.

Name	# Train Images Bona Fide (PA)	# Test Images Bona Fide (PA)	PA types	Sensors
LivDet 2009 [166]	1000 (1000) 1000 (1000)	1000 (1000) 1000 (1000)	Ecoflex, Gelatine, Latex, Modasil, WoodGlue	Biometrika Italdata
LivDet 2011 [257]	1000 (1000) 1000 (1000) 1000 (1000) 1000 (1000)	1000 (1000) 1000 (1000) 1000 (1000) 1000 (1000)	Gelatine, latex, PlayDoh, Silicone, Wood Glue, Ecoflex	Biometrika DigitalPersona ItalData Sagem
LivDet 2013 [86]	1000 (1000) 1000 (1000) 1250 (1000) 1250 (1000)	1000 (1000) 1000 (1000) 1250 (1000) 1250 (1000)	Body Double, Latex, PlayDoh, Wood Glue, Gelatine, Ecoflex, Modasil	Biometrika ItalData CrossMatch Swipe
LivDet 2015 [175]	1000 (1000) 1000 (1000) 1000 (1000) 1510 (1473)	1000 (1500) 1000 (1500) 1000 (1500) 1500 (1448)	Ecoflex, Gelatine, Latex, Liquid Ecoflex, RTV, WoodGlue, Body Double, PlayDoh, OOMOO	Biometrika DigitalPersona GreenBit CrossMatch
LivDet 2017 [176]	1000 (1200) 1000 (1200) 999 (1199)	1700 (2040) 1700 (2676) 1700 (2028)	Body Double, Ecoflex, Wood Glue, Gelatine, Latex, Liquid Ecoflex	GreenBit Orcanthus DigitalPersona
LivDet 2019 [186]	1000 (1200) 1000 (1200) 1000 (1000)	1020 (1224) 990 (1088) 1019 (1224)	Body Double, Ecoflex, Wood Glue, Gelatine, Latex, Liquid Ecoflex	GreenBit Orcanthus DigitalPersona
LivDet 2021 [39]	1250 (1500) 1250 (1500)	2050 (2460) 2050 (2460)	Latex, RProFast, Nex Mix 1, Body Double, Elmer's Glue, GLS20, RFast30	GreenBit Dermalog
MSU FPAD [41]	2250 (3000) 2250 (2250)	2250 (3000) 2250 (2250)	Ecoflex, PlayDoh, 2D Matte Paper, 2D Transparency	CrossMatch
MSU FPAD v2 [42]	4743 (4912)	1000 (leave-one-out)	2D Printed Paper, 3D Universal Targets, Conductive Ink on Paper, Dragon Skin, Gelatine, Gold Fingers, Latex Body Paint, Monster Liquid Latex, Play Doh, Silicone, Transparency, Wood Glue	CrossMatch

[1] The dataset release agreement for all LivDet databases can be found at https://livdet.org/registration.php.

[2] Similarly, the dataset release form for the MSU FPAD dataset can be found at http://biometrics.cse.msu.edu/Publications/Databases/MSU_FPAD/

neural networks to learn the decision boundary between bona fide and PA images. Some researchers have proposed training their algorithms on smaller patches of the fingerprint images as a way to deal with limited amounts of available training data, which roughly increases the number of training images by a factor proportional to the number of patches [41]. However, given the increased scrutiny over privacy concerns related to biometric datasets, it is not certain whether any PA fingerprint datasets will remain available in the future, motivating the need for synthetic data.

Another challenge related to limited training data is that of unseen PAs, or fingerprint images arising from never before seen PA instruments. This problem is also commonly referred to in the literature as cross-material generalization. Some strategies proposed to improve the cross-material performance of PA detectors include learning a tighter boundary around the bona fide class via one-class classifiers [58,72], incorporating adversarial representational learning to encourage robustness to varying material types [92, 193], or applying style transfer to mix textures from some known PA materials to better fill the space of unknown texture characteristics that may be encountered [44,79]. Similar ideas may apply to synthetic data generation, where new material types can be synthesized by mixing characteristics of known PAs.

6.2.2 Synthetic Fingerprint Generation

Research on synthetic fingerprint generation began in the early 2000s with the introduction of SFinGe [33]. Since then, many subsequent works have followed utilizing either hand-crafted approaches [125, 131, 268], learning-based approaches [6, 11, 19, 21, 23, 76, 171, 173], or a combination of both [71, 173, 252]. Classical methods, such as SFinGe, are useful for many applications due to the controllable nature of the generation process; however, they still lack the level of realism needed to close the domain gap to real fingerprint images. On the other hand, more recent learning-based approaches have generated increasingly realistic fingerprint ridge patterns but could not generate multiple impressions of the same finger. Fortunately, hybrid approaches, such as [71] and [252], incorporate domain knowledge into the learning-based generation process and can generate high quality fingerprints with multiple impressions per finger. Motivated by the success of hybrid approaches, we also employ a similar architecture as [71] to generate multiple, realistic

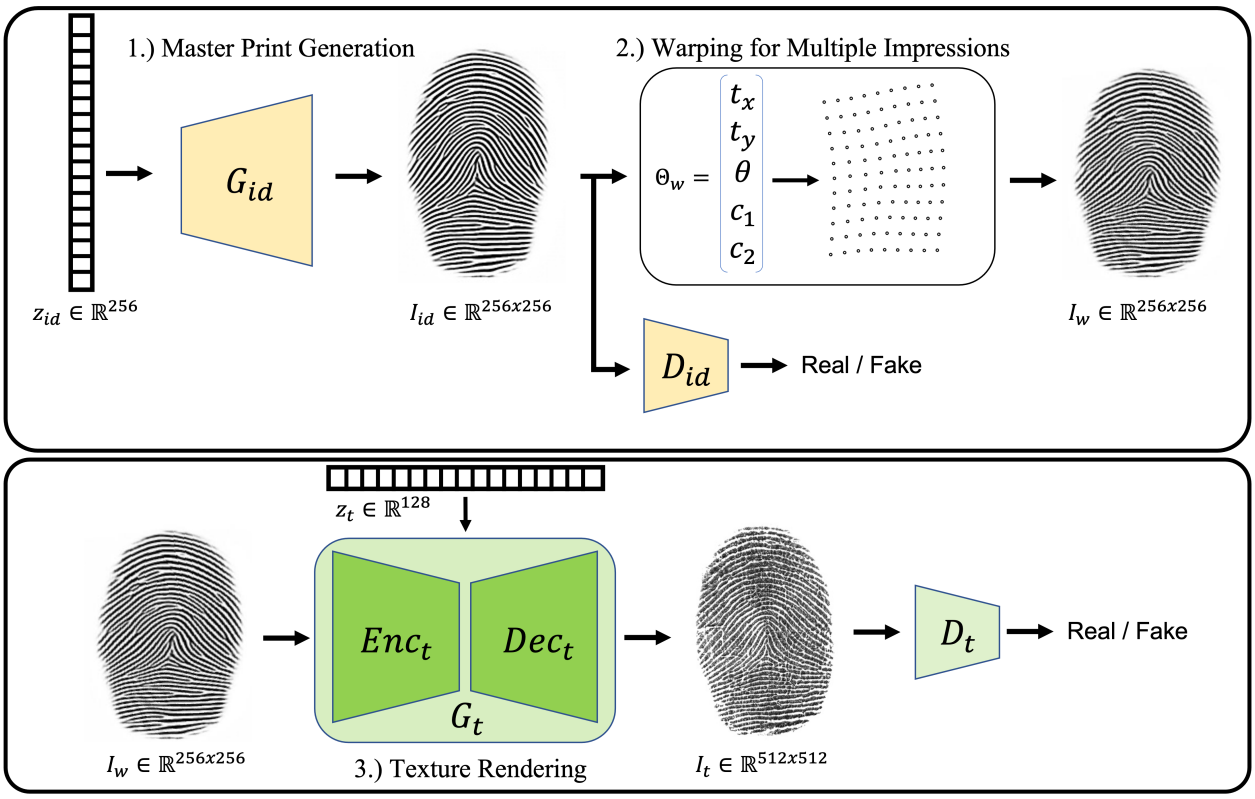


Figure 6.2 Overview of the SpoofGAN Architecture.

fingerprint images of each finger. However, unique to our approach, we also have the ability to simulate realistic PA impressions for each finger in a variety of different PA artifact "styles". To the best of our knowledge, this is the first study on synthetic PA fingerprint image generation.

6.3 Proposed Synthetic Presentation Attack Fingerprint Generator

In this section, we detail the process of generating synthetic bona fide and PA fingerprints. Motivated by the success of previous multi-stage fingerprint generation methods (e.g., [36, 71, 252]), SpoofGAN generates highly realistic fingerprints in multiple stages. First, unique fingers are synthesized through generating binary master fingerprints which define the fingerprint ridge structure of the finger. Following the master print synthesis stage, perturbations such as random rotation, translation, and non-linear deformation are applied to simulate realistic, repeat impressions. Finally, each generated fingerprint impression is input to a second neural network to impart realistic textures which mimic a database of real fingerprints. An overview of the entire process is given in Figure 6.2.

6.3.1 Master Print Synthesis

The first step in generating synthetic fingerprints with SpoofGAN is generating binary master fingerprints $I_{id} \in \{0,1\}^{256\times256}$ from a random vector $z_{id} \in \mathcal{R}^{256}$ sampled from a standard normal distribution (i.e., $z_{id} \sim \mathcal{N}(0,1)$). In particular, we used a standard BigGAN [22] architecture for this task, consisting of a generator G_{id} and a discriminator D_{id}. Since many PA impressions can exhibit non-realistic fingerprint ridge structures, either from artifacts introduced in the fabrication (e.g., bubbles in the ridges) or in the presentation process (e.g., smudges due to the high elasticity of some PA material types), we chose to train G_{id} using a database of only bona fide fingerprint impressions consisting of 38,164 images captured on a CrossMatch Guardian200 fingerprint reader. As we will show later, these artifacts for the PA impressions can be introduced in the later texture rendering stage of our synthesis pipeline. The network is trained via an adversarial loss shown in Equation 6.1, where I is a binary fingerprint image extracted from a real fingerprint image using the Verifinger v12.0 SDK.

$$\mathcal{L}_{adv}(G,D) = \mathbb{E}_I\left[logD(I)\right] + \mathbb{E}_z\left[log(1-D(G(z)))\right] \tag{6.1}$$

6.3.2 Generating Multiple Realistic Impressions

To generate multiple impressions from a single master print, we apply realistic rotation, translation, and non-linear deformation for each subsequent impression. Rotations are applied via a uniform random sampling in the range [-30°, 30°], whereas translations in both the x and y directions are uniformly sampled in the range [-25, 25] pixels. Finally, realistic non-linear deformations are applied via a learned, statistical deformation model proposed by Si et al. in [215]. The distortion parameters were learned from a database of 320 distorted fingerprint videos in which the minutiae locations in the first and last frames were manually labeled, and the displacements between corresponding minutiae points were used to estimate a distortion field via a Thin-Plate-Spline deformation model [20]. The distortion fields were condensed into a subset of eigenvectors, e_i, computed from a Principal Component Analysis of the covariance matrix estimated from the 320

videos. By varying the coefficients, c_i, multiplied to each of the eigenvectors, we vary the magnitude of distortion, d, applied to an input fingerprint along realistic distortion directions according to equation6.2, where λ_i are the eigenvalues of each eigenvector and $\bar{d}$ is the average distortion field. For our implementation, we randomly sample the coefficients of the two largest eigenvectors from a normal distribution with mean 0 and standard deviation of 0.66, which were empirically determined to produce reasonable distortions.

$$d \approx \bar{d} \sum_{i=1}^{t} c_i \sqrt{\lambda_i e_i} \tag{6.2}$$

6.3.3 Texture Rendering

The final stage of our fingerprint generation process consists of imparting each fingerprint with a realistic texture that mimics the distribution of real bona fide and PA images. For the generator, G_t, we use an encoder-decoder architecture which translates an input, warped binary image, I_w, into a realistic fingerprint impression, I_t. To promote diversity in the rendered images, a random texture vector sampled from a standard normal distribution (i.e., $z_t \sim \mathcal{N}(0, 1)$) is injected into the network and encoded into γ and β parameters for performing instance normalization on the intermediate feature maps of G_t. Finally, the discriminator, D_t utilizes the same architecture used in the binary master print synthesis network.

The goal of our texture renderer is two-fold: i.) generate realistic texture details and ii.) maintain the fingerprint ridge structure (i.e., identity) of the rendered fingerprint between corresponding impressions of the same finger. Thus, we introduce two losses in addition to the conventional GAN loss (eq. 6.3) to maintain the ridge structure of textured fingerprints. The first is an identity loss to minimize the L_2 distance between feature embeddings of corresponding fingerprint impressions using a SOTA fingerprint matcher DeepPrint [68] (eq. 6.4), and the other is an L_2 pixel loss between ground truth binary images and binary images extracted from the textured fingerprints (eg. 6.5). The L_2 pixel loss is computed on the binary images, rather than the grayscale images, to allow for the network to generate diverse "styles" in the generated fingerprints to simulate different

pressure, moisture content, and contrast in subsequent impressions; all of which would lead to slightly different loss values compared to the ground truth image unless first converted to binary ridge images. To make the process of binarization of the generated fingerprints differentiable, we train a convolutional autoencoder to binarize input fingerprints which is trained on 38,164 grayscale/binary image pairs. The overall losses for G_t and D_t are given in equations 6.6 and 6.7, respectively.

1. GAN loss: Classical min-max GAN loss between the discriminator, $D_t(\cdot)$, trying to classify each original fingerprint image, I, as real and each synthetic fingerprint $I_t = G_t(I_w)$ as fake. Meanwhile, $G_t(\cdot)$ is trying to fool $D_t(\cdot)$ into thinking its outputs come from the original image distribution.

$$\mathcal{L}_{adv} = \mathbb{E}_I\ [logD(I)] + \mathbb{E}_{I_w}\ [log(1 - D(G(I_w)))] \tag{6.3}$$

2. DeepPrint loss: L_2 distance between the DeepPrint embedding, R, extracted from the ground truth grayscale image and the DeepPrint embedding, $\hat{R}$, extracted from the synthesized grayscale fingerprint image.

$$L_{dp} = \frac{1}{2}\sum(R - \hat{R})^2 \tag{6.4}$$

3. Image/pixel loss: L_2 loss between the ground truth binary fingerprint image, I_w, and synthesized binary fingerprint image, $\hat{I_w}$.

$$L_i = \frac{1}{2}\sum_{x,y}(I_w(x,y) - \hat{I_w}(x,y))^2 \tag{6.5}$$

4. Overall loss for $G_t(\cdot)$: $\lambda_1 = 1$, $\lambda_2 = 2$, and $\lambda_3 = 10$ (determined empirically).

$$\mathcal{L}_{G_t} = \lambda_1\mathcal{L}_{adv} + \lambda_2\mathcal{L}_{dp} + \lambda_3\mathcal{L}_i \tag{6.6}$$

5. Overall loss for $D_t(\cdot)$:

$$\mathcal{L}_{D_t} = \mathcal{L}_{adv} \tag{6.7}$$

Unlike the binary master print synthesis and warping stages, an individual texture rendering network is trained for each material type (bona fide, ecoflex PA, PlayDoh PA, etc.). Due to the limited number of images in our PA dataset, we pretrained a texture rendering network on the

282K unique fingerprint database taken from the MSP longitudinal database introduced in [262]. Initially, following the pretraining, two texture rendering networks are trained further, one on the dataset of 38,164 bona fide only impressions and the other on the 3,366 PA fingerprint images consisting of all PA types aggregated together. Finally, we further finetune the model trained on all PAs for each of the individual PA types to give more fine-grained control on the specific PA style being generated. Thus, unlike the binary master print synthesis and warping stages which are shared, each individual PA type has it's own rendering network. Alternatively, a conditional GAN structure could be used to generate PA classes of each type within a single network; however, we found that due to the very limited amount of training images in some PA types (e.g., 50 images), finetuning for just a few epochs on each PA type individually produced higher quality images.

6.3.4 Training Details

For training the master print generator, G_{id}, an Adam optimizer with a learning rate of 0.0001 was used, whereas a Moving Average Optimizer with an initial learning rate of 0.0004 was used to train the discriminator, D_{id}. Furthermore, the master print generator was trained with a batch size of 8 on two NVIDIA GeForce RTX 2080 Ti GPUs for a total of 178 epochs, where each epoch contained 100 batches. To help balance the training, the generator was updated twice for every update of the discriminator. The architecture for both G_{id} and D_{id} are given in Table 6.2 and Table 6.3, respectively.

Finally, G_t and D_t of the texture renderer utilized the same optimizers as G_{id} and D_{id}, respectively; however, G_t was updated 3 times for every update of D_t. To increase the diversity in the generated samples, multiple checkpoints of the texture renderer are used in generating the synthetic data that was used for training the PAD model. The full architectures for G_t and D_t are given in Tables 6.4 and 6.5, respectively. For completeness, the architecture for the CNN-based fingerprint binarizer used in training the texture render is given in Table 6.6, and the architecture for the texture encoder (which encodes γ and β parameters from a random texture vector) utilized in G_t is given in Table 6.7.

This database is not publicly available, but the pretrained model can be made available upon request.

Table 6.2 Architecture for $G_{id}(\cdot)$ ($Ch = 48$).

Layer	Output Dim.
0. Input	512
1. ReLU(Dense)	$12,288$
2. Reshape	$4 \times 4 \times 16ch$
3. ResBlock Up†	$8 \times 8 \times 16ch$
4. ResBlock Up†	$16 \times 16 \times 8ch$
5. ResBlock Up†	$32 \times 32 \times 8ch$
6. ResBlock Up†	$64 \times 64 \times 4ch$
7. ResBlock Up†	$128 \times 128 \times 2ch$
8. Self Attention	$128 \times 128 \times 2ch$
9. ResBlock Up†	$256 \times 256 \times ch$
10. Tanh(Conv2d(Relu(Batch Norm))))	$256 \times 256 \times 1$

† Layer contains conditional batch norm.

Table 6.3 Architecture for $D_{id}(\cdot)$ ($Ch = 48$).

Layer	Output Dim.
0. Input	$256 \times 256 \times 1$
1. ResBlock Down	$128 \times 128 \times ch$
2. ResBlock Down	$64 \times 64 \times 2ch$
3. Self Attention	$64 \times 64 \times 2ch$
4. ResBlock Down	$32 \times 32 \times 4ch$
5. ResBlock Down	$16 \times 16 \times 8ch$
6. ResBlock Down	$8 \times 8 \times 8ch$
7. ResBlock Down	$4 \times 4 \times 16ch$
8. ResBlock	$4 \times 4 \times 16ch$
9. Dense(Global Sum Pooling(ReLU))	1

6.4 Experimental Results

In this section, we aim to validate the realism of our synthetic bona fide and PA images via several qualitative and quantitative experiments. First, we provide details on the datasets involved in the following experiments, followed by some example fingerprint images generated by SpoofGAN to qualitatively compare with real fingerprint images. Finally, several quantitative metrics are used to compare the utility and distribution of SpoofGAN generated fingerprint images compared to the real fingerprint images.

6.4.1 Datasets

A main motivation for this paper is the lack of large-scale, publicly available fingerprint PAD datasets. Some of the largest datasets that are available have resulted from the biennial LivDet competition series dating as far back as 2009 [39,86,175,176,186,257]. A more comprehensive list of the fingerprint PA datasets currently available to the research community is given in Table 6.1, whereas the datasets used in this paper are given in Table 6.8. In this paper we focus our experiments

Table 6.4 Architecture for $G_t(\cdot)$ ($Ch = 48$).

Layer	Output Dim.
0. Input	$256 \times 256 \times 1$
1. ReLU(Batch Norm(Conv2d))	$128 \times 128 \times 32$
2. ReLU(Batch Norm(Conv2d))	$64 \times 64 \times 64$
3. ReLU(Batch Norm(Conv2d))	$32 \times 32 \times 128$
4. ReLU(Batch Norm(Conv2d))	$16 \times 16 \times 256$
5. ReLU(Batch Norm(Conv2d))	$8 \times 8 \times 512$
6. Dense	512
7. Dense	$16ch$
8. Reshape	$4 \times 4 \times ch$
9. ResBlock Up†	$8 \times 8 \times 16ch$
10. ResBlock Up†	$16 \times 16 \times 8ch$
11. ResBlock Up†	$32 \times 32 \times 8ch$
12. ResBlock Up†	$64 \times 64 \times 4ch$
13. Self Attention	$64 \times 64 \times 4ch$
14. ResBlock Up†	$128 \times 128 \times 2ch$
15. ResBlock Up†	$256 \times 256 \times ch$
16. ResBlock Up†	$512 \times 512 \times ch$
17. Tanh(Conv2d(ReLU(Batch Norm)))	$512 \times 512 \times 1$

† Layer contains conditional batch norm.

Table 6.5 Architecture for $D_t(\cdot)$ ($Ch = 48$).

Layer	Output Dim.
0. Input	$512 \times 512 \times 1$
1. ResBlock Down	$256 \times 256 \times ch$
2. ResBlock Down	$128 \times 128 \times ch$
3. ResBlock Down	$64 \times 64 \times 2ch$
4. Self Attention	$64 \times 64 \times 2ch$
5. ResBlock Down	$32 \times 32 \times 4ch$
6. ResBlock Down	$16 \times 16 \times 8ch$
7. ResBlock Down	$8 \times 8 \times 8ch$
8. ResBlock Down	$4 \times 4 \times 16ch$
9. ResBlock	$4 \times 4 \times 16ch$
10. Dense(Global Sum Pooling(ReLU))	1

on fingerprint images obtained via the CrossMatch optical reader from LivDet 2013, LivDet 2015, and the Government Controlled Test (GCT) dataset of bona fide and PA fingerprints collected as part of the IARPA ODIN program. Our training dataset for SpoofGAN, referred to as GCT 1-5, consists of 38,164 bona fide fingerprint images and 3,366 PA fingerprint images from 2,007 fingers and 11 different PA types (Dragon Skin, Ecoflex, Paper, Silicone, Transparency, Gelatine, Glue, PDMS, Knox Gelatine, Gummy Overlay, and Tattoo). For our evaluations, we have followed the same train/test protocol referenced in LivDet2015 and LivDet2013, as well as reserved a fraction of the GCT dataset, GCT 6, as an evaluation dataset.

We selected CrossMatch for our experiments since it is one of the most popular slap (4-4-2) capture readers used in law enforcement, homeland security and civil registry applications.

Table 6.6 Architecture for Fingerprint Binarizer.

Layer	Output Dim.
0. Input	$256 \times 256 \times 1$
1. ReLU(Instance Norm(Conv2d)))	$128 \times 128 \times 64$
2. ReLU(Instance Norm(Conv2d)))	$64 \times 64 \times 128$
3. ReLU(Instance Norm(Conv2d)))	$32 \times 32 \times 256$
4. ReLU(Instance Norm(Conv2d)))	$16 \times 16 \times 384$
5. ReLU(Instance Norm(Conv2d)))	$8 \times 8 \times 512$
6. UpScale2d(ReLU(Layer Norm(Conv2d))))	$16 \times 16 \times 384$
7. UpScale2d(ReLU(Layer Norm(Conv2d))))	$32 \times 32 \times 256$
8. UpScale2d(ReLU(Layer Norm(Conv2d))))	$64 \times 64 \times 128$
9. UpScale2d(ReLU(Layer Norm(Conv2d))))	$128 \times 128 \times 2$
10. UpScale2d(ReLU(Layer Norm(Conv2d))))	$256 \times 256 \times 2$
11. Tanh(Layer Norm(Conv2d))))	$256 \times 256 \times 1$

Table 6.7 Architecture for the texture encoder used in the conditional batch norm layers of $G_t(\cdot)$.

Layer	Output Dim.
0. Input	128
1. ReLU(Dense)	128
2. ReLU(Dense)	128
3. ReLU(Dense)	128
4. ReLU(Dense)	128

6.4.2 Qualitative Analysis of Synthetic Bona Fide and PA Images

Example synthetic bona fide and PA fingerprints of varying material types (i.e., presentation Attack Instruments) are shown in column (b) of Figure 6.3 with corresponding real examples in column (a) shown for reference. Visually, SpoofGAN is generating PA examples that closely resemble the target material type, which can be seen in the different texture characteristics for each material seen in the synthetic examples (e.g., Paper vs. Tattoo, etc.). Additionally, looking across the rows of the synthetic SpoofGAN images, we notice that the underlying fingerprint ridge structure is successfully being preserved across the different PA styles being generated. Figure 6.4 further highlights SpoofGAN's ability to successfully generate multiple impressions of each finger in both bona fide and PA styles.

6.4.3 Quantitative Analysis of Synthetic Bona Fide and PA Images

Similar to previous works on synthetic fingerprint generation [11, 71], we have evaluated the quality of SpoofGAN fingerprints on several quantitative metrics, including PAD performance by a pretrained PA detector, the distribution of minutiae count, type, and quality extracted from SpoofGAN generated fingerprints compared to a real fingerprint dataset, NFIQ2 quality scores,

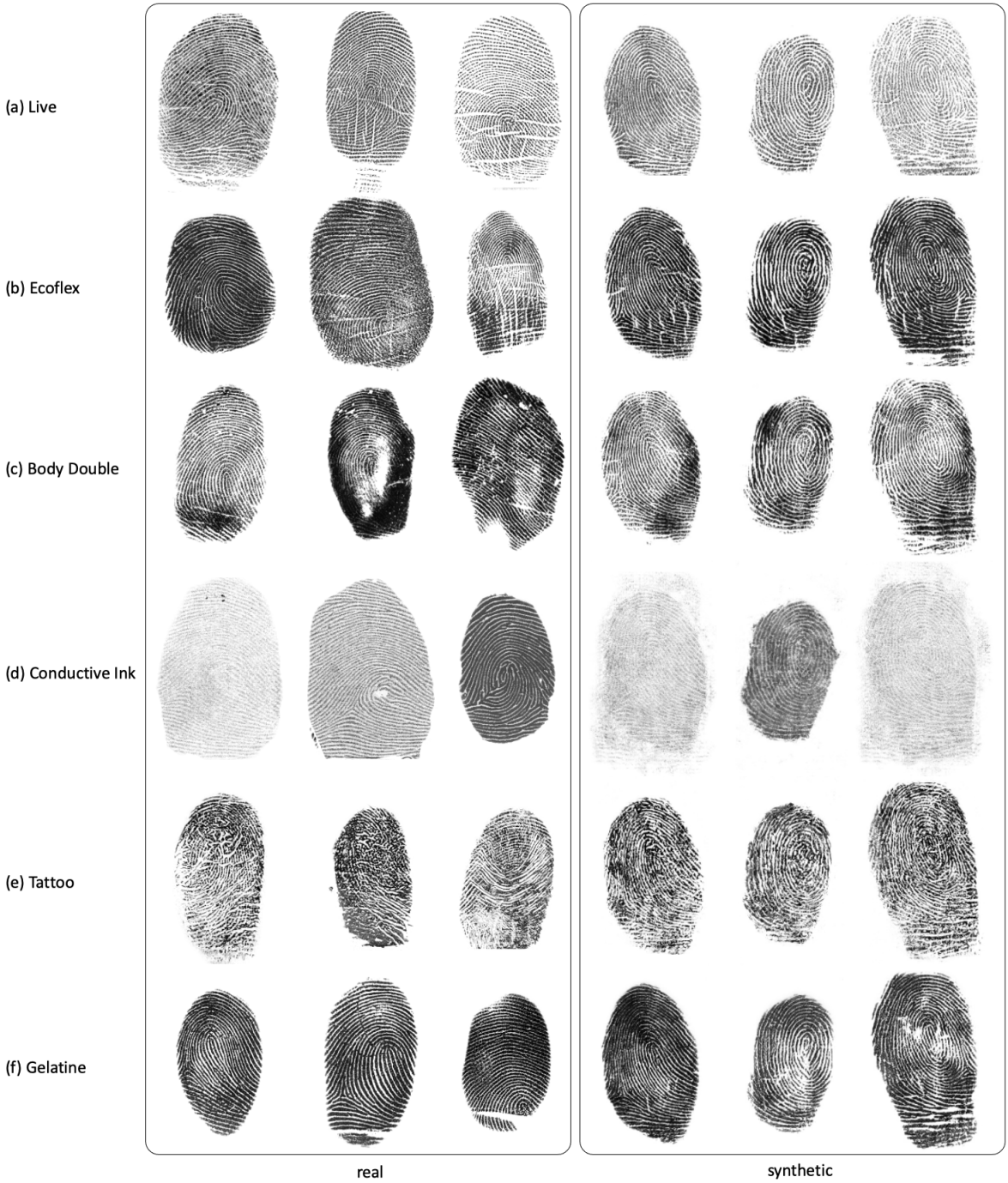


Figure 6.3 Example real bona fide and PA images and synthetic bona fide and PA images generated by SpoofGAN of various material types: (a) bona fide, (b) ecoflex, (c) body double, (d) conductive ink on paper, (e) tattoo, and (f) gelatine.

Table 6.8 Summary of the PA datasets used in our experiments.

Dataset	LivDet 2013	LivDet 2015	GCT 1-5	GCT 6	MSU FPADv2
Fingerprint Reader	CrossMatch	CrossMatch	CrossMatch	CrossMatch	CrossMatch
Model	L Scan Guardian	L Scan Guardian	Guardian200	Guardian200	Guardian200
Resolution (dpi)	500	500	500	500	500
# Bona Fide Images (Train/Test)	1,250 / 1,250	1,510 / 1,500	38,164 / 0	7,357 / 14,236	4,743 / 1,000
# PA Images (Train/Test)	500 / 440	1,473 / 1,448	3,366 / 0	2,550 / 1,829	4,912[†]
PA Materials	Body Double, Latex, PlayDoh, Wood Glue	Ecoflex, Gelatine, PlayDoh, OOMOO Body Double	Dragon Skin, Ecoflex, Knox Gelatine, Silicone, Transparency, Gelatine, Glue, PDMS, Tattoo Paper, Gummy Overlay	Ecoflex, Silicone, Gummy Overlay, Tattoo, Knox Gelatine	Paper, Transparency, 3D Unviersal Targets, Conductive Ink on Paper, Dragon Skin, Gelatin, Gold Fingers, PlayDoh, Latex Body Paint, Monster Liquid Latex, Silicone, Wood Glue

[†] There are 4,912 total PA images but the number of train/test images depends on which PA is left-out for the cross-material generalization evaluation.

match score distributions from a SOTA fingerprint matcher, and identity leakage experiments.

6.4.3.1 Presentation Attack Detection Performance of Real vs. Synthetic Fingerprints

Our first evaluation to verify the quality of our synthetic bona fide and PA fingerprints is to see whether a pretrained PAD algorithm trained on similar, real fingerprints performs equally well on our synthetic fingerprints. In particular, we pretrained an Inception v3 network on the GCT 1-5 data to classify between bona fide and PA fingerprint samples. Then, we evaluated the PAD performance on LivDet 2015 CrossMatch images compared to an equivalent sized database of synthetic fingerprints. As shown in Table 6.9, the attack presentation classification error rate (APCER), computed at threshold corresponding to bona fide classification error rate (BPCER) of 0.2%, is similar across multiple PA types of both datasets, supporting our hypothesis that the synthetic samples should be useful in training additional PAD models without access to a large

Figure 6.4 Example images of multiple impressions of the same finger generated by SpoofGAN. (a) and (b) show three impressions each of two fingers rendered in a bona fide style, whereas (c) and (d) show three impressions each of the same two fingers in a PA style (ecoflex and body double, respectively).

Table 6.9 PAD model trained on real fingerprints (GCT 1-5) and evaluated on LivDet 2015 CrossMatch images (top row) and an equivalent-sized synthetic fingerprint dataset (bottom row)[1]. Results given in APCER at a threshold corresponding to BPCER=0.2%.

	BodyDouble	**Ecoflex**	**PlayDoh**	**OOMOO**	**Gelatine**
Real	0%	0.43%	1.42%	0.42%	0%
Synthetic	0%	0%	0%	0.42%	0.88%

[1] There are 1,500 bona fide and 1,448 PA fingerprint test images for CrossMatch in LivDet 2015, which we have replicated with synthetic data. Specifically, the PA images consist of 300 Body Double, 270 Ecoflex, 297 OOMOO, 281 PlayDoh, and 300 Gelatine images.

database of real bona fide and PA fingerprints for training. Lastly, the embeddings of both real and synthetic images in the T-SNE embedding space suggest high similarity between the embeddings of real and synthetic images (see Figure 7.5).

APCER and BPCER are the standard metrics according to ISO/IEC 30107-1:2016; however, other metrics have also been reported in the literature, including true detection rate at a fixed false detection rate, where BPCER=FDR and APCER=1-TDR (albeit, APCER is computed per PAI).

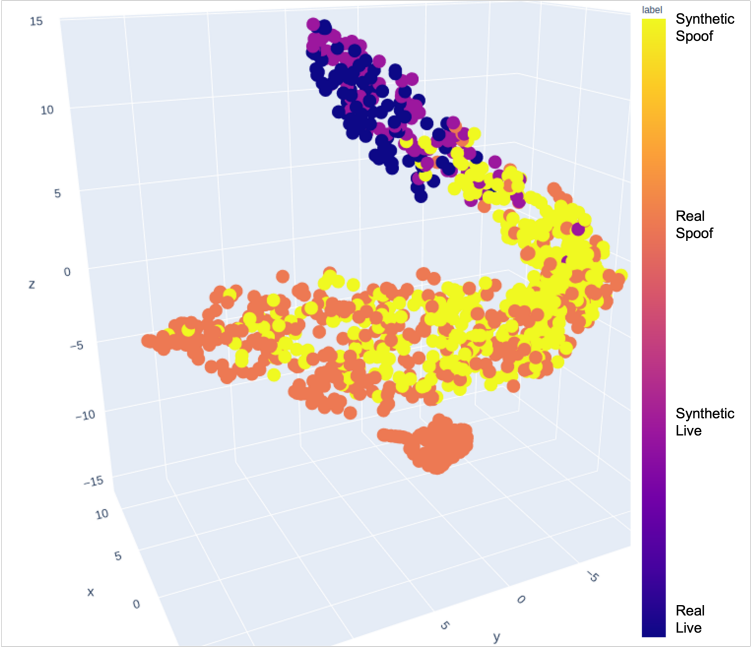


Figure 6.5 3D visualization of 2,048-dimensional embeddings of real bona fide and PA images from the LivDet2015 CrossMatch dataset compared with embeddings of our synthetic bona fide and PA images generated from SpoofGAN. Best viewed in color.

6.4.3.2 Feature Similarity Between Real and Synthetic Fingerprints

For synthetic fingerprint images to be useful as a substitute for real fingerprint images, the features between a database of real fingerprint images and synthetic fingerprint images should closely align. For this analysis, we computed several statistics from the LivDet 2015 CrossMatch training dataset (bona fides only) and 1,500 SpoofGAN bona fide fingerprint images which are shown in Table 6.10. In terms of fingerprint area, SpoofGAN images are, on average, smaller compared to the real fingerprint database. Since our training dataset consists of images of all 10 fingers, there is a bias toward smaller fingerprints considering the thumb as a minority class. Given this assumption, it is perhaps unsurprising that a GAN-based generation approach might exaggerate this class imbalance and generate smaller fingerprint area impressions. This problem is related to mode-collapse and has been noted in several GAN related works [89, 205, 222], with some recent papers proposing strategies to improve the generation process in class-imbalanced datasets [114, 168].

Next, we computed the NFIQ 2.0 quality metric [228] on both datasets (see Figure 7.8). The

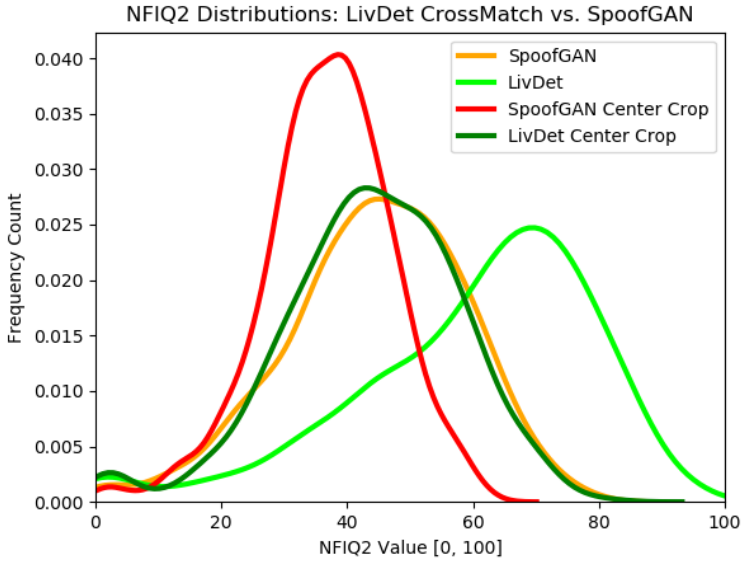


Figure 6.6 NFIQ2 quality scores for LivDet 2015 CrossMatch bona fide images and SpoofGAN synthetic bona fide images. Due to smaller fingerprint area of SpoofGAN images compared to LivDet CrossMatch images, the NFIQ2 distributions of the full images is quite different between the two datasets; however, if we compute NFIQ2 scores on 256×256 center crops, the means of the distributions are much closer.

NFIQ 2.0 scores for SpoofGAN are, on average, lower compared to LivDet. However, since one of the features considered in NFIQ 2.0 is the fingerprint area, we recomputed the scores on a 256×256 center crop of each of the fingerprints and observed that independent of fingerprint area, the NFIQ scores between SpoofGAN images and LivDet are much more aligned (36.88 ± 10.18 vs. 43.65 ± 14.12).

Lastly, we computed some additional metrics specific to the distribution of minutiae since many of the state-of-the-art fingerprint algorithms incorporate minutiae information. The average minutiae count and minutiae quality computed by Verifinger 12.0 are given in Table 6.10. The average number of minutiae found in SpoofGAN seems to be lower compared to the CrossMatch images from LivDet 2015; however, the minutiae per Megapixel is similar for both datasets (59.74 vs. 59.49). The minutiae quality given by Verifinger is also very similar between the two datasets (71.89 vs. 70.78).

Table 6.10 Metrics for real and SpoofGAN fingerprint images. Minutiae quality and NFIQ2 scores have a range of [0, 100].

	LivDet 2015 CrossMatch		SpoofGAN	
Measure	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
Total Minutiae Count	55.56	18.43	40.45	11.57
Ridge Ending Minutiae Count	30.58	12.12	20.99	6.67
Ridge Bifurcation Minutiae Count	24.98	9.18	19.46	6.61
Verifinger Minutiae Quality	71.89	16.00	70.78	15.15
Fingerprint Area (Megapixels)	0.93	0.30	0.68	0.15
Fingerprint Image Quality (NFIQ2)	60.18	19.35	44.34	14.38

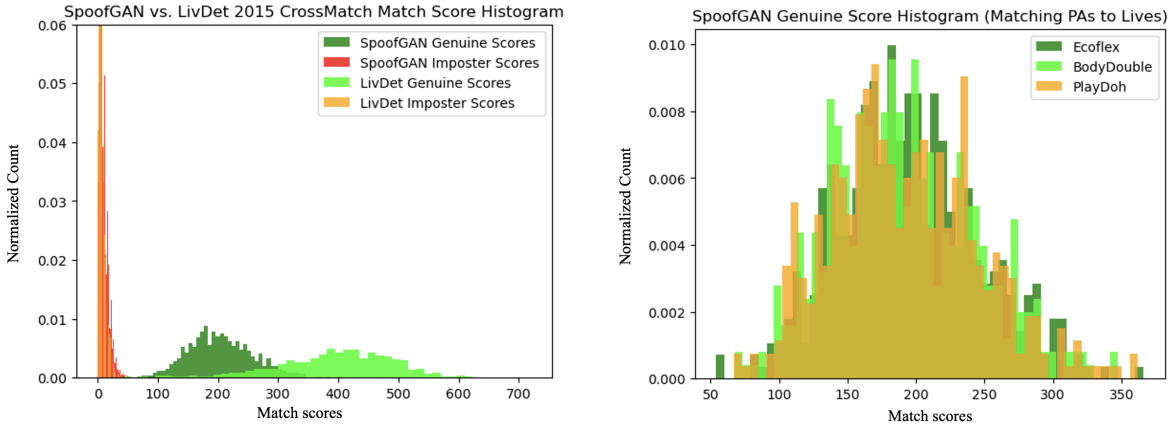
6.4.3.3 Diversity in the Generated Fingers

To verify that SpoofGAN generated fingerprints mimic the similarity score distribution of real bona fide and PA fingerprints, we have computed genuine and imposter matches with Verifinger v12.0. In particular, we computed match scores (genuine and imposter) between the bona fide impressions of the LivDet 2015 CrossMatch images as well as between the bona fide samples of synthetic SpoofGAN fingerprints. These distributions are shown in Figure 6.7 (a). This figure highlights that SpoofGAN is generating diverse fingers with similar intraclass and interclass variation as the real LivDet 2015 CrossMatch dataset, albeit producing slightly lower genuine scores compared to the real dataset. However, in terms of recognition performance at a fixed false acceptance rate (FAR), the performance between the two datasets is quite similar, despite the slightly shifted genuine distribution of the SpoofGAN images (see Table 6.11).

Furthermore, we computed genuine score distributions between the individual PA types generated by SpoofGAN and their corresponding bona fide impressions. These distributions are given in Figure 6.7 (b). Here we see that Verifinger is able to successfully match PA and bona fide fingerprint images belonging to the same finger, which we believe opens the door to synthesising a large-scale PA and recognition dataset that can be used to train and evaluate joint PAD and recognition algorithms.

6.4.3.4 Identity Leakage

A major advantage of generating synthetic fingerprint data is that, theoretically, no fingerprint ridge structure matches that of an actual user in the training database. However, there remains a concern that synthesis methods, such as GANs, may inadvertently over-fit and leak private



(a) (b)

Figure 6.7 Verifinger 12.0 match score distributions of the real LivDet 2015 CrossMatch L Scan Guardian database vs. an equivalent sized synthetic SpoofGAN database. (a) match scores computed between the bona fide impressions of each dataset and (b) match scores computed between bona fide and PA impressions for the SpoofGAN images.

Table 6.11 TAR at Varying FAR Thresholds for LivDet 2015 CrossMatch Test Data vs. SpoofGAN Images.

FAR	**LivDet**	**SpoofGAN**
0.01% @ threshold=48	99.87%	100%
0.001% @ threshold=60	99.80%	100%
0.0001% @ threshold=72	99.74%	99.80%
1e-05% @ threshold=84	99.41%	99.47%
1e-06% @ threshold=96	99.35%	98.87%

information from the training corpus [15]. Therefore, it is instructive to investigate whether, and to what degree, any of our SpoofGAN generated images are revealing, i.e., match with sufficient confidence, the identities present in our training database. Toward this end, we have computed match scores between 1,500 SpoofGAN generated bona fide fingers and each of the 38,164 real bona fide fingers in our training set. Out of the roughly 57.2 million $(1,500 \times 38,164)$ potential matches, only 50 comparisons exceeded the matching threshold of 48 set by Verifinger for a false acceptance rate of 0.01% with a maximum match score of 81. Furthermore, the 50 matches resulted from just 29 SpoofGAN generated fingers out of the 1,500 evaluated. Some example matched SpoofGAN and real fingerprint image pairs are shown in Figure 6.8 along with their corresponding match scores.

As part of future work, the risk of identity leakage could be further mitigated with either

Figure 6.8 Example SpoofGAN and real training database image pairs with corresponding match scores given by Verifinger. For each pair, the left image is a SpoofGAN fingerprint image and the right image is a real fingerprint image. The threshold for a genuine match at an FAR of 0.01% for Verifinger is 48, indicating that the identity of these training images have been "leaked" by SpoofGAN in the corresponding generated images.

i.) a larger training database to avoid the network from simply memorizing training samples, ii.) performing an identity check with the training database upon generation of each fingerprint to trigger a re-sampling if matched with an existing finger (though potentially an expensive operation), or iii.) composing the master print pattern using SFinGe or other classical method (albeit, at the expense of realism in the generated friction ridge pattern).

6.4.4 Improved Presentation Attack Detection with Synthetic Fingerprints

Ultimately, our synthetic fingerprints should offer some utility in advancing the training of fingerprint PAD algorithms. Toward this end, we have augmented three existing, publicly available datasets of bona fide and PA fingerprints with our synthetic fingerprints in an effort to improve the performance beyond that achievable when training on the real bona fide and PA images from each dataset alone. For this evaluation, we have trained several PAD models on the following training set compositions: i.) synthetic bona fide and PA images only, ii.) real bona fide and PA images only, iii.) synthetic bona fide and PA images plus only real bona fide images, and iv.) synthetic bona fide and PA images plus real bona fide and PA images. We used the SpoofBuster model, which consists of two Inception v3 networks, one trained on the whole image input and the other trained on 96×96 minutiae centered patches [41]. The final PA score is the weighted fusion of the two networks, with

a minutiae patch score weight of 0.8 and a whole image score weight of 0.2. Each of the models are trained from scratch using Tensorflow on a single Nividia TitanX 1080 GPU on their respective datasets with identical hyper-parameters (learning rate of 0.01, step decay learning rate schedule, Adam optimizer with default parameters, and total training updates of 200,000 steps).

Shown in Table 6.12, we have evaluated each of the models on their respective test sets. Despite the lower performance when training on synthetic data alone compared to training on real data, we see improvement in the overall PA classification performance when the real training data is augmented with samples from our synthetic bona fide and PA generator. For example, the error of the minutiae patch model trained on real data from LivDet 2013 is reduced by 91.03% (from 15.60% to 1.40%) when augmented with synthetic data. Similarly, the error on LivDet 2015 is reduced from 0.48% to 0.0%, where the error on GCT 6 remained the same at 0.0%.

Interestingly, as seen in Table 6.13, we also see substantial improvements in cross-material generalization when incorporating synthetic SpoofGAN images into the training dataset of our PAD model. Specifically, we compared the cross-material generalization of a PAD model trained on LivDet 2013 with the same PAD model trained on LivDet 2013 augmented with SpoofGAN images. We observe drastic reduction in APCER across 7 different PA types from the MSU FPADv2 dataset which are not included in LivDet 2013 training dataset, nor were these PA types replicated in the synthetic images obtained from SpoofGAN that were used to augment the training dataset. Overall, the average APCER reduced from 76.71% to 54.03% due to the addition of SpoofGAN training images.

Furthermore, Figure 6.9 shows the trend in performance on LivDet 2015 as we keep the number of synthetic training samples fixed but varying the percentage of real data included when training the whole image-based PAD model. This figure suggests that when augmenting the training set with synthetic data, just 25% of the original (real) data is required to obtain similar performance to training on 100% of the real data alone, which significantly reduces the time and resources required for data collection to obtain similar performance. In fact, the behavior of the real training data curve (shown in blue) in the early stages (e.g., 5% and 10% of the total training data) exhibits a very sharp

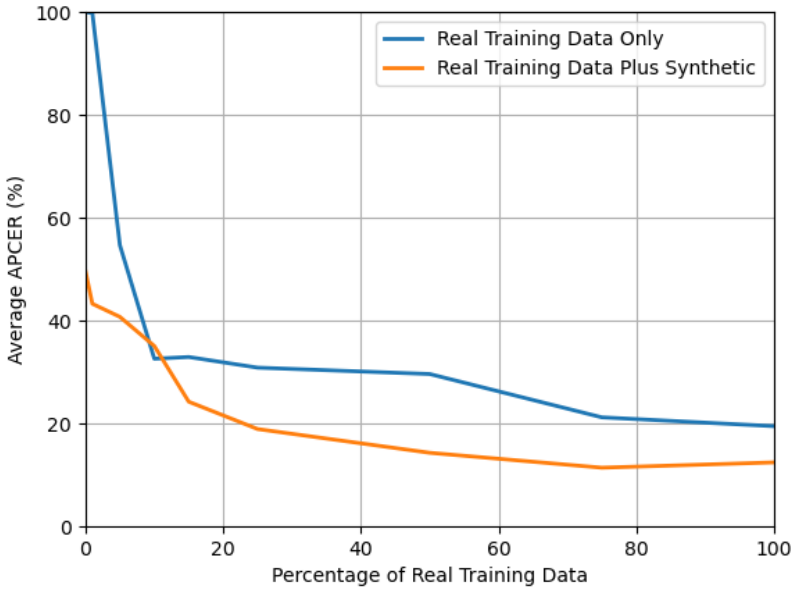


Figure 6.9 PAD performance as we keep the number of synthetic training samples fixed and vary the percentage of real data included in training. Performance reported as average APCER @ BPCER = 0.2% across all PA types in the LivDet 2015 dataset.

decrease in APCER, suggesting that many of the PA vs. bona fide features can be learned from very few samples; however, the subtle features that give the PAD model the last 10% of improvement require 3 or 4 times as many data samples. Importantly, from the real plus synthetic curve (shown in orange), the additional data required to gain that extra 10% can instead be generated by SpoofGAN.

Lastly, even if researchers and practitioners have access to a large, private database of real bona fide fingerprint images, collecting an equivalent database of PA fingerprints is more difficult and costly; therefore, they may wish to augment their database of real bona fide images with synthetic PA images. As seen in Table 6.12, mixing synthetic data with only real bona fide fingerprint examples improves the performance over training on just the synthetic examples alone; however, synthetic PA data is still not a substitute for collecting real PA examples as the performance still lags quite significantly.

6.5 Conclusion and Future Work

In this chapter, we presented a GAN-based synthesis method for generating high quality 512 × 512 plain fingerprint impressions of both bona fide and PA varieties. We demonstrated the utility of

Table 6.12 PAD performance when trained on various combinations of real and synthetic data. We have followed the established protocols of train/test split for each of the evaluation datasets, which are shown in Table 6.8

Training Data Composition				Average APCER @ 0.2% BPCER		
Real Bona Fide	**Real PA**	**Synthetic Bona Fide**	**Synthetic PA**	**LivDet2015**	**LivDet2013**	**GCT6**
		✓	✓	63.47%	57.70%	43.92%
✓		✓	✓	19.89%	27.00%	13.14%
✓	✓			0.48%	15.60%	**0.0%**
✓	✓	✓	✓	**0.0%**	**1.40%**	**0.0%**

Table 6.13 Cross-material PAD performance on unseen material types from the MSU FPADv2 dataset, comparing a PAD model trained on only LivDet 2013 vs. training with LivDet 2013 plus synthetic data from SpoofGAN. Results reported as APCER @ 0.2% BPCER.

Training Dataset	**3D Universal Targets**	**Conductive Ink on Paper**	**Dragon Skin**	**Gold Fingers**	**2D Paper**	**Silicone**	**Transp-arency**	**Average ± Std. Dev.**
LivDet2013[1]	82.50%	100.0%	79.65%	43.05%	95.84%	36.69%	99.27%	76.71 ± 0.26%
LivDet2013 + SpoofGAN[2]	65.00%	64.00%	70.53%	34.92%	64.03%	15.51%	64.23%	54.03 ± 0.21%

[1] Trained on bona fide and PAs from LivDet 2013.
[2] Trained on bona fide and PAs from LivDet 2013 plus synthetic bona fide and PAs (of the same material types as LivDet 2013) from SpoofGAN.

our synthetically generated PA fingerprints in improving the performance of a PA detector beyond that achieved when training on real PA fingerprint datasets only, both in terms of seen PAs and cross-material generalization. Additionally, our synthetic fingerprints closely resemble a database of real fingerprints both qualitatively and quantitatively in terms of various statistics, such as distribution of minutiae, NFIQ 2.0 image quality, PA classification, and match score distributions computed by the state-of-the-art Verifinger 12.0 SDK. Finally, since our method is capable of generating multiple genuine and imposter fingerprints of unique fingers in both bona fide and PA types, we open the door for large-scale training and evaluation of joint fingerprint PAD and recognition algorithms, overcoming a current limitation given the existing scale of publicly available PA fingerprint datasets.

Despite the demonstrated utility of our synthetic PA images, there remains several limitations that will be addressed in future work. First, given the lower standard deviation across the various fingerprint metrics presented in Table 6.10, the diversity of the generated images could be improved. A related issue, specific to fingerprint PAD, is the ever increasing novelty of PA materials and types that may be encountered in the future; thus, instilling the generation process with the ability to adapt to novel PA types is a promising future research direction. Lastly, training the synthetic fingerprint

generator in an online fashion with a PAD network may provide additional supervision to generate more useful bona fide and PA examples to improve the PAD performance. In this next chapter, we extend our synthetic fingerprint generation capabilities toward universal fingerprint generation to address the limitations in intra-class diversity present in SpoofGAN and other existing fingerprint generation methods.

CHAPTER 7

UNIVERSAL FINGERPRINT GENERATION

The utilization of synthetic data to train fingerprint recognition models has garnered increased attention in the biometric field due to its potential to alleviate privacy concerns surrounding sensitive biometric data. However, current methods for generating fingerprints have limitations in creating varied impressions of the same finger with useful intra-class variations. To tackle this challenge, this chapter presents *GenPrint*, a framework designed to produce fingerprint images of various types while maintaining identity and offering humanly understandable control over different appearance factors such as fingerprint class, acquisition type, sensor device, and quality level. Unlike previous fingerprint generation approaches, GenPrint is not confined to replicating style characteristics from the training dataset alone: it enables the generation of novel sensor and style attributes from unseen fingerprint acquisition devices during inference without requiring additional fine-tuning. To accomplish these objectives, we developed GenPrint using latent diffusion models with multimodal conditions (text and image) for consistent generation of style and identity. Our experiments leverage a variety of publicly available datasets for training and evaluation. Results demonstrate the benefits of GenPrint in terms of identity preservation, explainable control, and universality of generated images. Importantly, the GenPrint-generated images yield comparable or even superior accuracy to models trained solely on real data and further enhances performance when augmenting the diversity of existing real fingerprint datasets.

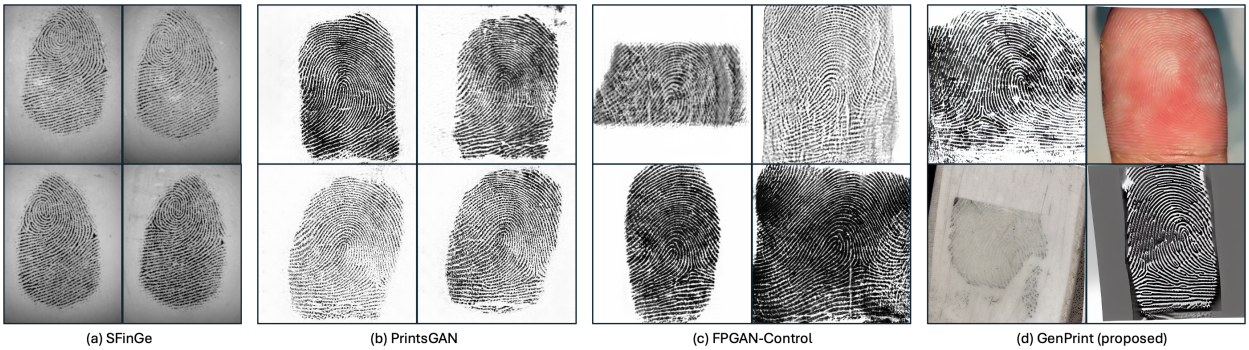


Figure 7.1 Synthetic fingerprint images generated by various baseline methods and the proposed GenPrint. The four images in each panel are impressions of the same finger to show case the intra-class variance of each method.

7.1 Introduction

The use of Artificial Intelligence Generated Content (AIGC) over the last few years has exploded due to advancements in model architectures and larger computation and data being used to train Generative AI (GAI) models [32]. In particular, text generation models, such as ChatGPT, have catapulted the field of GAI into the public view since its public release in November of 2022 [184]. Following in its wake came stunning advancements in image and video generation models, such as ImageGen [90] and SORA [185], utilizing denoising diffusion probabilistic model (DDPM) frameworks. Since then, DDPM models have proliferated as the center of attention in many top computer vision conferences and journals. Notably, their probabilistic framework and straightforward optimization process makes DDPMs more stable and easy to train compared to generative adversarial networks (GANs) [89], one of the predominant frameworks for image generation previously. Furthermore, the work of Dhariwal and Nichol further demonstrated the advantages of diffusion models over GANs for image generation in terms of image quality [57]. Indeed, the introduction of GANs by Goodfellow et al. [89] in 2014 and the recent surge in DDPM models have revolutionized GenAI capabilities across enumerable industries and applications.

Artificial fingerprint generation is one application which has received increased interest for the potential of synthetic data for training and evaluation of algorithms, aided by recent privacy and ethical concerns as well as difficulty and cost associated with collecting biometric data. Before the explosion of deep learning techniques, fingerprint generation methods began with intelligent, hand-crafted methods to simulate convincing fingerprint patterns and textures [33]. Importantly, these methods allowed for generating multiple images of the same finger, opening the door to training and evaluation of fingerprint recognition algorithms.

Early GAN-based methods drastically improved the realism of the generated prints but lacked control over the fingerprint identity being generated [11, 23, 76, 171, 173, 198]. Subsequent works aimed to fill this gap by replacing each stage of the mutli-stage generation pipeline of hand-crafted methods with GANs, preserving the identity of the generated fingerprints at each stage [71, 96, 253]. However, with the exception of identity, other appearance factors remained obscured and

uncontrollable, such as the specific fingerprint class (e.g., arch, loop, and whorl), acquisition type (e.g., rolled, slap, contactless, swipe, and latent), sensor characteristics (e.g., optical, capacitive, thermal, etc.), and quality level (e.g., high, average, and low) of the generated prints. Shoshan et al. [214] proposed FPGAN-Control to disentangle identity and appearance factors in the latent space and allowed for swapping between different appearance latent vectors to achieve some degree of control over intra-class variations (e.g., acquisition type, sensor, and pressure level); however, this method lacked explicit, humanly explainable control over appearance factors.

Recent advancements in text to image generation models utilizing DDPMs have demonstrated very realistic and controlled image generation capabilities. In this chapter, we aim to leverage DDPM advancements for controllable fingerprint image generation utilizing multimodal conditions (text and image) for improved generation capabilities. We leverage text prompts to allow for guidance of explainable appearance factors and rely on image style embeddings for factors not easily expressed in language. Importantly, an added benefit of our novel image style condition is that the generation outputs are no longer constrained to interpolating between the domain of the seen training data, for it allows for zero-shot generation of novel fingerprint sensor characteristics not seen during training. For a visual comparison, figure 7.1 shows some example synthetic images generated from SFinGe, PrintsGAN, FPGAN-Control and the proposed model which we refer to as GenPrint. The four images in each panel are of impressions of the same finger identity to showcase the intra-class variance of each method, which demonstrates the improved diversity which GenPrint is capable of generating.

More concisely, the contributions of this research are the following:

1. A controllable latent diffusion model, GenPrint, using text and image conditions for highly realistic and diverse synthetic fingerprint generation.
2. GenPrint is capable of generating fingerprints of any acquisition type, sensor, fingerprint class, and quality, including fingerprint styles not seen during training without any additional fine-tuning (e.g., zero-shot fingerprint style generation).
3. The generation process is controllable (both in appearance and identity preservation) and

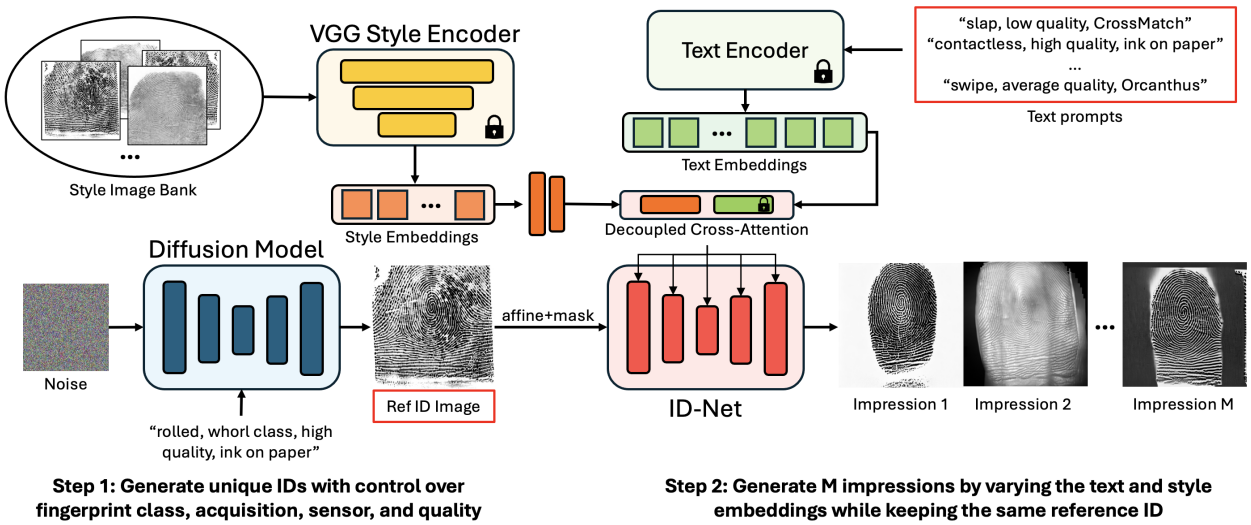


Figure 7.2 Architecture of GenPrint.

explainable with humanly interpretable text prompts.

4. The utility of GenPrint synthetic images is validated through experiments showcasing improved recognition performance of models trained on GenPrint images compared to real datasets and other benchmark fingerprint generation methods.
5. We also demonstrate the utility of GenPrint images for evaluating fingerprint recognition systems by replacing real data for large-scale identification experiments.
6. Open-sourcing a dataset of 100K synthetic finger identities with 15 impressions of various acquisition devices to the research community.

7.2 Related Work

7.2.1 Hand-crafted Fingerprint Generation Methods

The seminal work of Cappelli et al. [33] utilized a combination of an elliptical shape generation model, mathematical ridge flow and Gabor filters for ridge pattern generation, and noise and distortion models to simulate realistic fingerprint patterns. Importantly, this model allowed for generating multiple impressions of the same finger leading to its adoption for aiding in training and evaluation of fingerprint recognition models. Despite its impressive capabilities and intelligent design, SFinGe is limited in its intra-class variations it is able to generate due to its hand-crafted nature (see subfigure (a) of Figure 7.1 for examples). More recent methods have turned to deep

learning techniques, starting with GANs, to learn the subtle intra-class variations that have led to more varied and realistic fingerprint images.

7.2.2 Fingerprint Generation via GANs

The introduction of GANs gave way to more realistic fingerprint generation that captured more realistic texture characteristics that are difficult to hand-design [11, 23, 76, 171, 173, 198]; however, early uses of GANs lacked control over the fingerprint identity being generated - severely limiting the utility of the generated fingerprints. Wyzykowski et al. [253], aimed to fill this gap by adopting CycleGAN as a wrapper around SFinGe generated images to impart them with more realistic textures, while leveraging SFinGe's ability to generate multiple impressions. However, the intra-class and inter-class variations were still limited by the hand-designed generation of SFinGe. Engelsma et al. went one step further and designed a multi-stage GAN method for generating highly realistic fingerprint ridge patterns with multiple impressions per finger and showed substantial improvement over SFinGe in utility for recognition model training [71]. Finally, Shoshan et al. [214] adopted a mixed variational autoencoder (VAE) and GAN architecture called FPGAN-Control to interpolate between latent identity and appearance vectors to be able to render fingerprint images in multiple different appearances. Still, this model lacked explicit control over the appearance factors, and the possible space of generated fingerprint styles is constrained to the distribution of styles belonging to the original training set.

7.2.3 DDPMs for Fingerprint Generation

To the best of our knowledge, DDPMs have only just begun to be investigated for artificial fingerprint generation [138, 232]. Tang et al. applied a vanilla DDPM to synthesize unconditional fingerprint patches and validated the realism compared to real fingerprint patches using the Fréchet Inception Distance (FID) metric [232]. Li and Yang also applied an unconditional DDPM model trained on a dataset of latent, rolled, and plain (i.e., slap) fingerprint images to randomly generate fingerprint impressions of these types [138]. They demonstrated the realism of the DDPM generated fingerprint images both in terms of NFIQ quantitative values and t-SNE qualitative comparisons to the real fingerprint images. However, their model lacked control over both the identity and

appearance of the generated fingerprints, which is critical for training and evaluation of fingerprint recognition models. To the best of our knowledge, the proposed GenPrint model is the first use of DDPMs for fingerprint generation with explicit control over both the identity and appearance of generated fingerprint images.

7.3 GenPrint: Controllable Multimodal Fingerprint Diffusion Model

GenPrint is a multimodal latent diffusion model [199] finetuned for fingerprint generation from a pretrained Stable Diffusion v1.5 model with weights made available from the Diffusers library [242]. In this section, we first describe the text to image fingerprint generation capabilities including the dataset curation process and fine-tuning procedure. Next, we describe the architectural design for incorporating style image embeddings into the Stable Diffusion pipeline and explain the zero-shot style generation capability it facilitates. Finally, the identity preservation process is described along with a detailed description of the full pipeline for generating synthetic fingerprint images with GenPrint. An overview of the architecture design is given in Figure 7.2.

7.3.1 Control Factors via Text Conditions

The first step in fine-tuning Stable Diffusion for text to fingerprint generation is obtaining a large corpus of fingerprint images and associated text descriptions. For this purpose, we aggregated data from multiple fingerprint datasets from predominately publicly available sources. These training datasets are listed in Table 7.1 along with the acquisition label, sensor label, and number of images for each dataset. Our aggregated dataset consists of data from five different acquisition types (rolled, slap, swipe, contactless, and latent) and thirty different sensing devices ranging from optical readers as well as capacitive, thermal, contactless, and latent surfaces.

Missing from many fingerprint datasets are annotations for fingerprint class (whorl, plain arch, tented arch, left loop, and right loop) and quality (low, average, and high quality), which are needed to impart the generator with this kind of control. To obtain these labels, we utilized Verifinger SDK v12.4 to extract class and NFIQ 2.0 [228] quality estimations. Since the NFIQ 2.0 metric was optimized for slap impressions utilizing frustrated total internal reflection (FTIR) optical imaging,

https://www.neurotechnology.com/verifinger.html

the quality levels across each acquisition type may vary distinctly. Thus, we fit individual quality distributions according to a normal distribution using images belonging to each acquisition category and assigned low, average, and high quality labels to image clusters based on the mean ± standard deviation.

Using these annotations we constructed text prompt labels for each training image utilizing the following template: "a {acquisition} fingerprint image, {class} pattern, {quality} quality, {sensor}, {sensing}", where the acquisition type is one of {rolled, slap, swipe, contactless, latent}, class is one of {whorl, plain arch, tented arch, left loop, right loop}, quality is one of {low, average, high}, sensor is one of the thirty training sensors listed in Table 7.1, and sensing type is one of {FTIR optical, direct-view optical, multispectral optical, capacitive, thermal}.

For fine-tuning Stable Diffusion on our text to fingerprint image dataset, we utilize the low-rank adaptation (LoRA) strategy for more efficient training with a rank of 128 [113]. The LoRA weights are finetuned with a learning rate of 0.0001, cosine scheduler [150], default Adam optimizer [130], and batch size of 96 spread across 8 Nvidia A100 GPUs. The model is trained for 500,000 steps and trained on fingerprint images of a resolution of 512×512 pixels.

7.3.2 Zero-shot Style Generation

Motivated by the fact that many of the textural intra-class variations present in fingerprint images are not easily expressed in language via simple text prompts, we turned toward a deep learning-based representation to capture those characteristics. In particular, we take a pretrained VGG [216] model trained on ImageNet to embed style embeddings for each training image. These style embeddings are injected into the diffusion model via cross-attention layers which are decoupled from the cross-attention layers from the textual embeddings used to control the explainable style factors. Our choice of VGG embeddings for style representation is motivated from two key insights: i.) the previous use of VGG for neural style transfer [124] and ii.) visualizing the separation of VGG style embeddings for various fingerprint sensor types in the t-SNE embedding space (see Figure 7.5).

During inference, style embeddings from various sensor types present in the training data can

Table 7.1 Training datasets for GenPrint.

Train Dataset	Acquisition Types	Sensor Types	No. Images (Fingers)
NIST SD14 [246]	Rolled	Ink on paper	54,000 (27,000)
FVC 2002 [155]	Slap	Desktop Scanner, TouchChip, DF90	2,400 (100)
FVC 2004 [156]	Slap, Swipe	CrossMatch, Digital Persona, Fingerchip	2,400 (100)
PLUS-MSL-FP [132]	Slap, Swipe	Eikon, Integrated Biometrics Columbo, Integrated Biometrics Curve, Lumidigm, Next Biometrics, Suprema RealScan G1, Digital Persona	106,712 (580)
MSU Infant Fingerprint [118]	Slap	SilkID	9,683 (1,921)
NIST SD302 (N2N) [78]	Slap, Contactless, Rolled	CrossMatch, Eikon, GreenBit, ANDI, S120, MorphoWave, DactyScan, LIVETOUCH, Futronic, RaspiReader	45,072 (1,600)
NIST SD302 Latent [78]	Rolled, Latent	Ink on paper, crime scene	7,586 (1,019)
MSP Latent [262]	Rolled, Latent	Ink on paper, crime scene	1,866 (933)
IIITD SLF [211]	Slap, Latent	CrossMatch, crime scene	480 (150)
MOLF [212]	Slap, Latent	Lumidigm, Secugen, CrossMatch, crime scene	65,512 (1,000)
MUST [159]	Slap, Latent	CrossMatch, crime scene	20,247 (120)
IIT Bombay Touchless and Touch-based [17]	Slap, Contactless	eNBioScan, smartphone	3,200 (200)
ISPFDv2 [158]	Slap, Contactless	Secugen, smartphone	57,600 (304)
UWA Benchmark 3D Fingerprint [270]	Slap, Contactless	CrossMatch, 3D scanner	18,266 (1,500)
ZJU Finger Photo and Touch-based Fingerprint [93]	Slap, Contactless	Digital Persona, smartphone	39,580 (824)

be sampled to generate images of that sensor. On the other-hand, even style embeddings extracted from images of a completely new, unseen sensor can be used to generate images in that new sensor domain. Therefore, our method is generalizable and allows for “zero-shot" fingerprint style generations without any additional fine-tuning required. This fact is later supported by empirical evidence in section 7.4.3 to produce new fingerprint characteristics of latent, optical, capacitive, and contactless sensors outside those seen during training.

7.3.3 Fingerprint Identity Preservation

Several strategies for identity preservation and personalization in diffusion models have been proposed. Some of these techniques, such as Textual Inversion [80] and DreamBooth [207], require additional fine-tuning for each new concept, whereas others, such as IP-Adapter [245] and PhotoMaker [141], can produce identity consistent generations for multiple subjects without inference time fine-tuning. Both IP-Adapter and PhotoMaker embed the identity of an input reference image or images into the diffusion process via cross-attention layers. This guides the diffusion model to generate images which are identity consistent with the input reference images. Empirically, we tried IP-Adapter but found that it lacked the fine-grained spatial control needed to maintain the fingerprint ridge structure throughout the image. To solve this, we turned to ControlNet [266], which is another adaptation to the diffusion model process in which reference images are provided to the diffusion model to guide the generation with spatially consistent outputs.

For fingerprints, the identity discriminative features which are consistent across multiple different acquisition and sensor types are the silhouettes of the ridge flow patterns giving rise to the relative orientation of minutiae points of each finger. We posit that ControlNet is a suitable choice for imparting our DDPM model with identity preservation. Therefore, we propose to adapt the ControlNet framework to provide explicit spatial consistency of the generated fingerprint ridge pattern by pre-pending a ridge extraction module to the input of our identity preserving diffusion model, ID-Net. This ridge extractor removes sensor dependent and other style characteristics from the input fingerprint control image leaving only the ridge pattern silhouette image to guide the spatial preservation of the fingerprint identity, including the location and orientation of minutiae

points. This, combined with the text and style embeddings providing the style information, allows our ID-Net to generate varying textural characteristics while maintaining the input fingerprint ridge pattern. The architecture for our ridge extraction model is the light-weight SqueezeUNet model, which has been successfully applied previously for fingerprint ridge extraction [98].

7.3.4 Generation Pipeline for GenPrint

The full generation pipeline for GenPrint consists of two stages. First, our finetuned stable diffusion model is used to generate full (i.e., rolled) fingerprint images of various fingerprint classes from a random noise vector. For this stage, the text prompt guiding the generation follows the template of “a rolled fingerprint image, {class} pattern", high quality, ink on stock paper", where the fingerprint class is randomly selected from the five available classes. This provides a full fingerprint ridge pattern for use in the subsequent generation stage which imparts controllable style variations to generate large intra-class variations. By varying the noise vector for each generation, completely new and unique fingerprint patterns are generated. This fact is supported in section 7.4.6, showcasing the inter-class separation of the generated fingerprints, and section 7.4.7, highlighting the low similarity between generated identities and the training fingerprint identities. In the second stage, the generated fingerprint images from the first stage are passed through ID-Net and imparted with varying appearances based on the style embeddings (from reference images either belonging to the training set or from new example images from unseen sensors) and different text prompts providing explainable acquisition, sensor, and quality factors.

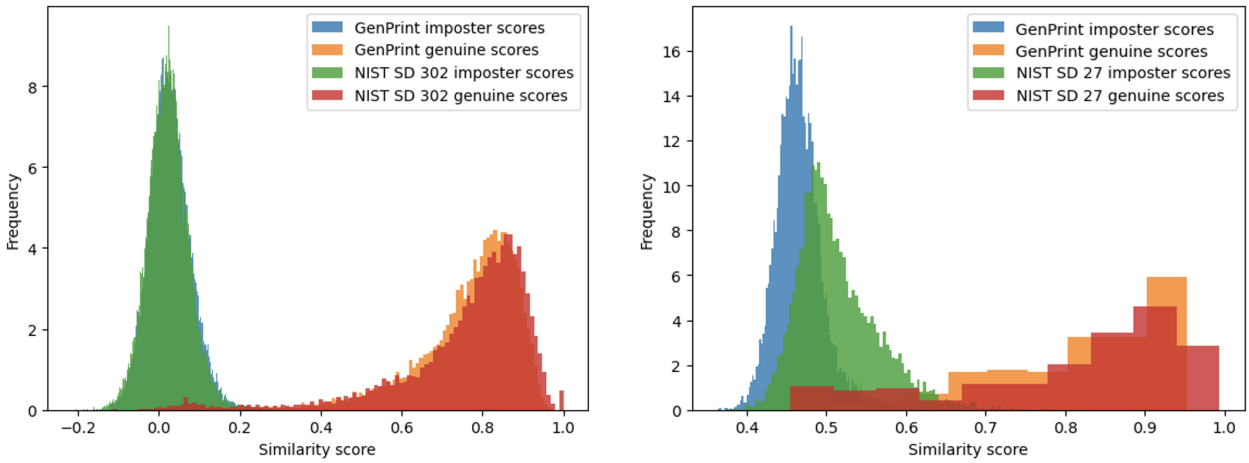
One critical observation we noticed was that the ControlNet framework was indeed very successful at constraining the local spatial details to be preserved in the generated images; however, often we found that the generated images had the tendency to over-constrain the generation process to preserve every detail of the input control image. This is undesirable if, for example, the desired output image is a slap fingerprint image where the input control image is a full, rolled fingerprint ridge pattern. The result is an unrealistic image with the full rolled fingerprint pattern in the style of the specified “slap" sensor input. Therefore, we apply a mask to the output of the ridge extractor aligned with the input text prompt to apply a realistic foreground mask for the specified acquisition

type. For example, if the prompt is to produce a slap fingerprint image, then an extracted mask of the fingerprint foreground area from one of the slap training images is applied to the input rolled fingerprint ridge pattern to produce an image with a fingerprint area resembling a realistic slap fingerprint. If instead the prompt is to generate a latent fingerprint, then a mask from a training latent fingerprint image is applied to the input control image to produce a realistic looking latent fingerprint with occluded areas of the ridge pattern.

Similarly, the ControlNet aspect of the ID-Net model will not produce realistic non-linear distortions to the generated images because it would modify the input fingerprint pattern supplied as the ControlNet input. Therefore, we also randomly sample realistic distortion grids to apply to the ControlNet image for each generation. These realistic distortion grids are obtained by computing minutiae displacements between genuine fingerprint pairs within the training dataset. During inference, an example distortion grid, indexed by the specified fingerprint acquisition type, is sampled and applied to the input reference image.

7.4 Experimental Results

In this section, we first evaluate the realism of GenPrint-generated images compared to real fingerprint images and other baseline fingerprint generation methods. We then verify the validity of each of the explainable control factors which GenPrint is trained to generate, including control over the fingerprint class, acquisition, sensor, and quality. Next, we examine GenPrint's adaptability for zero-shot style generation by using GenPrint to generate fingerprint images following the style characteristics of the unseen Latent Fingerprint in the Wild (LFIW) dataset consisting of three new latent fingerprint types, one optical sensor, one capacitive sensor, and one contactless sensor [148]. Additionally, we evaluate the utility of GenPrint-generated images for training fingerprint recognition models and compare it with other fingerprint generation methods both when training on only synthetic images and for augmenting a set of real fingerprint images with additional synthetic fingerprint impressions. Furthermore, we show the potential for GenPrint images to be used for evaluation of fingerprint recognition models as a replacement for real fingerprint images in large-scale identification experiments. Next, we verify the uniqueness and independence of



(a) NIST SD302 and similar size and composition GenPrint Dataset

(b) NIST SD27 and similar size and composition GenPrint Dataset

Figure 7.3 AFR-Net similarity score distributions for (a) NIST SD302 and similar GenPrint dataset and (b) NIST SD27 and similar GenPrint latent dataset.

GenPrint-generated finger identities compared to the set of training fingerprint identities from which it was trained. Finally, we include a discussion on the failure cases and limitations of GenPrint.

7.4.1 Realism of Generated Fingerprints

To validate the realism of GenPrint-generated fingerprints, we performed two experiments: i.) comparing the genuine and imposter score distribution of GenPrint images to a similar composition of real fingerprint images and ii.) comparing various fingerprint and minutiae related statistics between real images and GenPrint synthetic images.

For the first experiment, we generated 400 unique synthetic fingers with 12 impressions each across a random selection of slap, rolled, and contactless fingerprints using GenPrint to mimic the size and sensor distribution of a test split of the NIST SD302 dataset consisting of 400 real finger identities with roughly 12 impressions each and a mix of different sensor and acquisition types. We then computed genuine (same identity) and imposter (different identity) score distributions using a pretrained AFR-Net [97] fingerprint recognition model for both the GenPrint-generated dataset and the test split of NIST SD302. The results are shown in subfigure (a) of Figure 7.3. Similarly, we repeated the experiment by generating an equivalent size and composition (258 unique latent and

rolled fingerprint pairs) to the NIST SD27 dataset and compared the score distributions in subfigure (b) of Figure 7.3. We chose these two real datasets as comparison because they encompass many of the different acquisition types (rolled, slap, contactless, and latent) that GenPrint is trained to generate. The realism of GenPrint-generated images is evident from the overlap in the distributions compared to both real fingerprint datasets. The recognition performance of each of the datasets is also very similar. For NIST SD302 and the corresponding GenPrint dataset, the true accept rate (TAR) at a false accept rate (FAR) of 0.01% is 96.33% and 97.33%, respectively. Similarly, for NIST SD27 and corresponding GenPrint latent dataset, the TAR is 63.57% and 68.33%, respectively.

Next, we compare various fingerprint statistics from 1,000 real, rolled fingerprint impressions from the NIST SD4 dataset to 1,000 synthetic rolled impressions generated by GenPrint and the baseline method PrintsGAN. The specific metrics being compared are summarized in Table 7.2 and include fingerprint area, minutiae count, and average quality of minutiae. Compared to the real fingerprint dataset, GenPrint and PrintsGAN images differ slightly between average fingerprint area compared to the real images, where PrintsGAN tends to produce smaller fingerprints and GenPrint tends to produce larger fingerprints. To normalize for the relative differences in fingerprint area, we computed the minutiae statistics on a center crop of 256×256 pixels. Compared to the real images, GenPrint images exhibit a higher degree of similarity than PrintsGAN images in terms of average minutiae count and quality. For example, GenPrint differs from the real fingerprint images in average minutiae count by 5.27, whereas PrintsGAN differs by 6.52. Similarly, GenPrint differs in minutiae quality by 0.02, whereas PrintsGAN differs by 1.15.

7.4.2 Consistency of Control Factors

In this section, we evaluate all the different explicit control factors which GenPrint is trained to accommodate via text prompts, including control over the fingerprint class, acquisition, sensor, and quality level.

7.4.2.1 Fingerprint Class

GenPrint is able to generate fingerprints of any of the five major classes of fingers: whorl, left loop, right loop, plain arch, and tented arch. Examples of each of the categories generated by

Table 7.2 Fingerprint statistics comparison of GenPrint and PrintsGAN generated images to real fingerprint images.

	MSP [262] (real dataset)	PrintsGAN	GenPrint
Minutiae count	37.18 ± 9.75	30.66 ± 6.98	42.45 ± 8.52
Minutiae quality	80.38 ± 10.36	81.53 ± 9.52	80.40 ± 9.83
Area (pixels)	192,285 ± 34,368	175,460 ± 25,189	211,599 ± 19,241

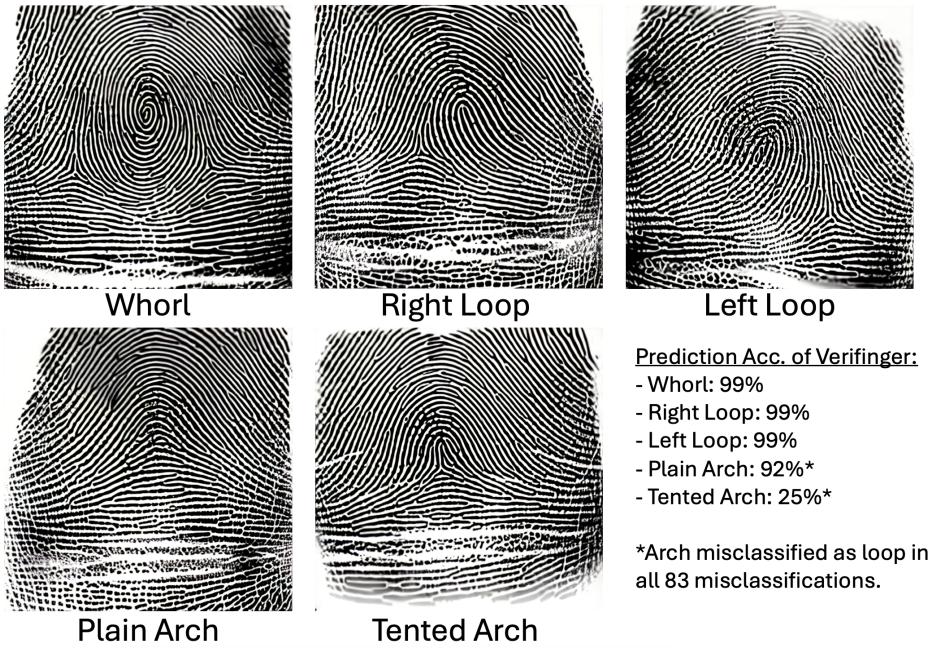
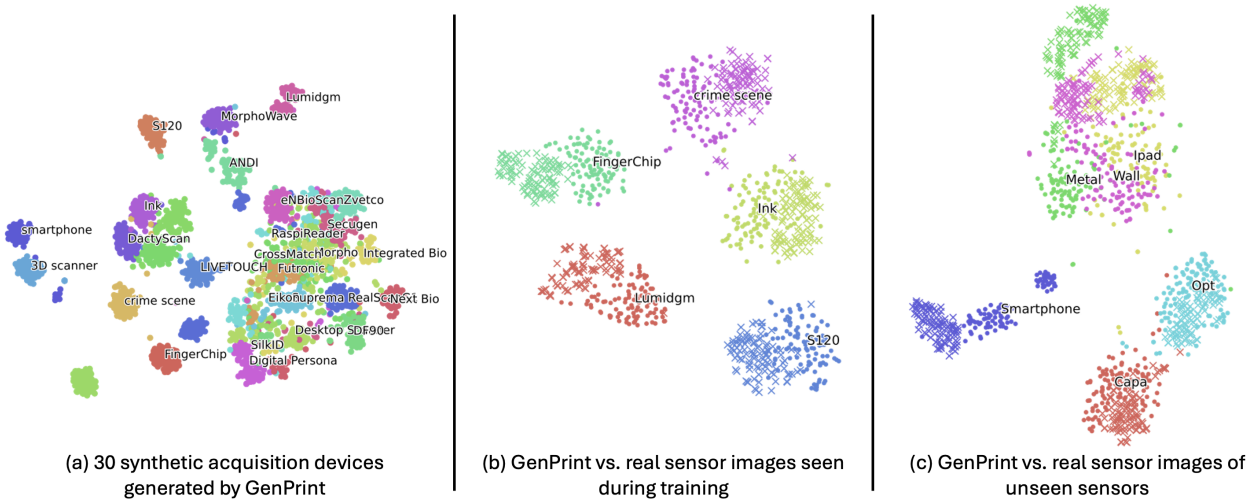


Figure 7.4 Example GenPrint images of different fingerprint classes and corresponding classification accuracy of Verifinger v12.4 SDK.



(a) 30 synthetic acquisition devices generated by GenPrint

(b) GenPrint vs. real sensor images seen during training

(c) GenPrint vs. real sensor images of unseen sensors

Figure 7.5 T-SNE plots to show (a) separation of GenPrint-generated images from different acquisition devices, (b) similarity of GenPrint images and corresponding real images of the same acquisition device, (c) similarity of zero-shot generated images to corresponding real images of novel acquisition devices which were not included in the training set of GenPrint.

GenPrint are shown in Figure 7.4. The consistency of GenPrint-generated images in following the fingerprint class prompt provided by the user is validated quantitatively using the commercially available fingerprint recognition software, Verifinger SDK v12.4. Specifically, we generate 100 unique finger identities using GenPrint in each of the five different fingerprint classes and classify each of the fingerprints using Verifinger and compute the accuracy between the Verifinger predictions and the ground truth class assigned by the input text prompts. The classification accuracy for whorl, left loop, and right loop fingerprints was 99%, indicating that 99 out of 100 generated fingerprints were classified by Verifinger as the same class intended to be generated by GenPrint. It turned out that the classification accuracy for Verifinger on the plain arch (92%) and tented arch types (25%) was much more challenging for Verifinger, which often misclassified the arch type as either left or right loop in all the misclassifications. Understandably, these two fingerprint classes can be difficult to distinguish given the similarity in the ridge patterns.

7.4.2.2 Fingerprint Acquisition and Sensor Type

GenPrint is trained on data from 30 different acquisition devices which consist of various rolled, slap, swipe, contactless, and latent fingerprint acquisition types. Some example images from different devices are given in Figure 7.6 along with corresponding GenPrint-generated images

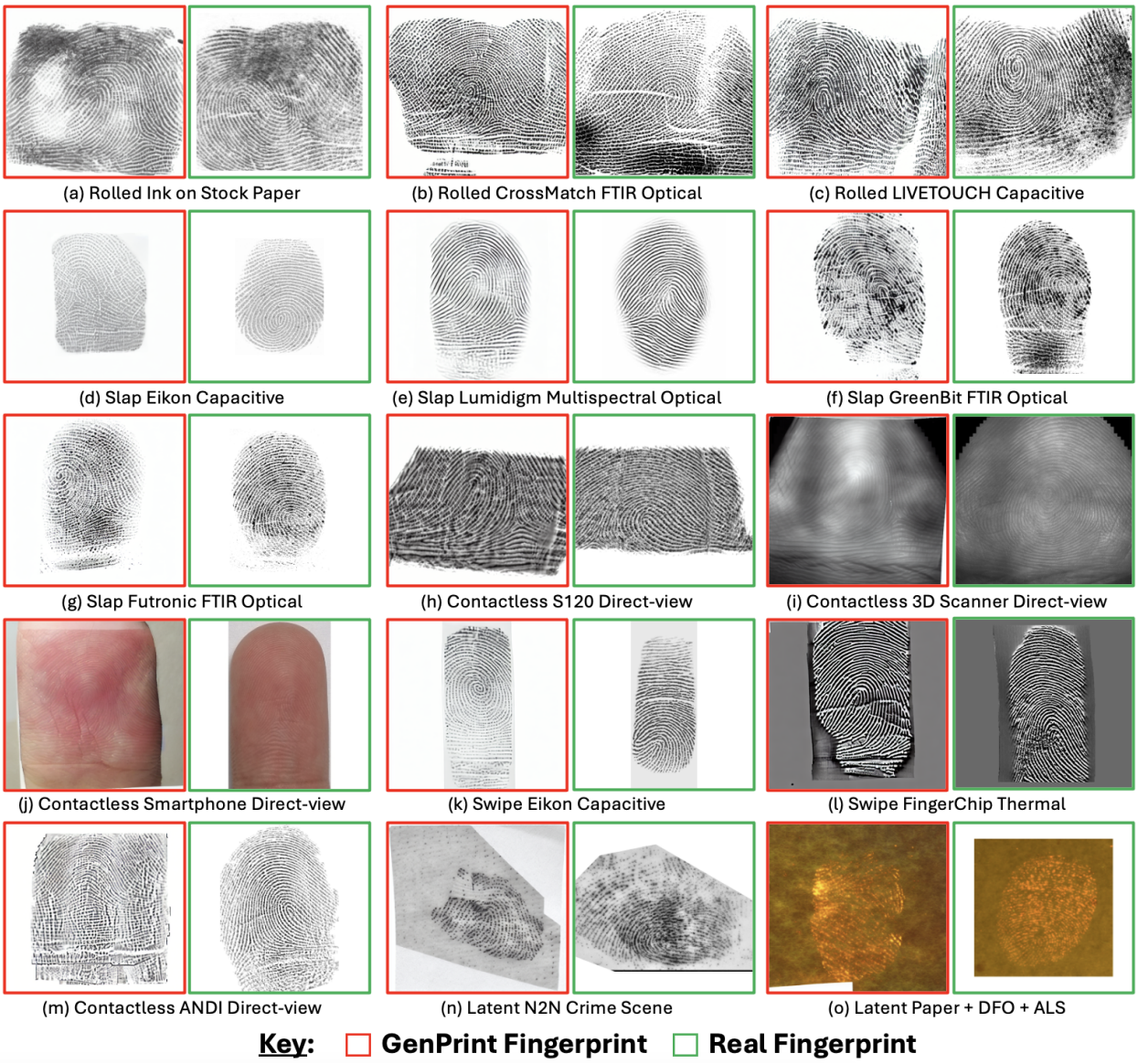


Figure 7.6 Example GenPrint-generated and real fingerprint images from corresponding acquisition device domains. In each pair, the left image is generated by GenPrint and the right image is a real fingerprint image of the same acquisition device to show the similarity of GenPrint images to real images with corresponding sensor characteristics.

in those same device domains, where the left image in each pair is a synthetic fingerprint generated by GenPrint, and the right image is an example fingerprint image from a real fingerprint database. Comparing GenPrint images and corresponding real images in the same sensor and acquisition types highlights the realism and diversity in the possible generation space of GenPrint.

To visualize the separability of all 30 acquisition device characteristics that GenPrint is trained to generate, we first generated 100 example fingerprint images in each acquisition device domain. Then, we extracted representation embeddings using a pretrained VGG network and plotted them in t-SNE [153] embedding space. The result is shown in subfigure (a) of Figure 7.5 which shows clear separation between very distinct acquisition devices and some small overlap in similar sensors, such as the large number of different slap FTIR optical devices sharing similar characteristics. Furthermore, we also generated VGG embeddings for 100 real fingerprint image examples in 5 different acquisition devices and embedded them into the t-SNE space along with their corresponding generated images from GenPrint to show the similarity between corresponding real and synthetic images of the same acquisition device domains.

7.4.2.3 Quality Control

There are two ways in which GenPrint can manipulate the quality of the generated images. The first is through the text prompt where the user can specify either low, average, or high quality, and the other is through passing a reference style image with a relatively low, average, or high quality appearance. Empirically, we found both approaches to work well. For validating the quality control of GenPrint, we generated datasets of 100 unique synthetic finger identities with 300 impressions of each of the five different acquisition types (rolled, slap, swipe, contactless, and latent) and used the text prompt to generate 100 of those impressions for each quality level (low, average, and high). Example low, average, and high quality images for three generated rolled fingerprints are given in Figure 7.7 for visualization. We then computed the NFIQ 2.0 quality score using the Verifinger SDK and plotted the quality distributions in Figure 7.8. There is clear separation among each of the quality levels across each of the acquisition types, verifying GenPrint's appropriate control over the quality of the generated fingerprints.

Figure 7.7 Example low, average, and high quality rolled fingerprint impressions of a finger generated by GenPrint.

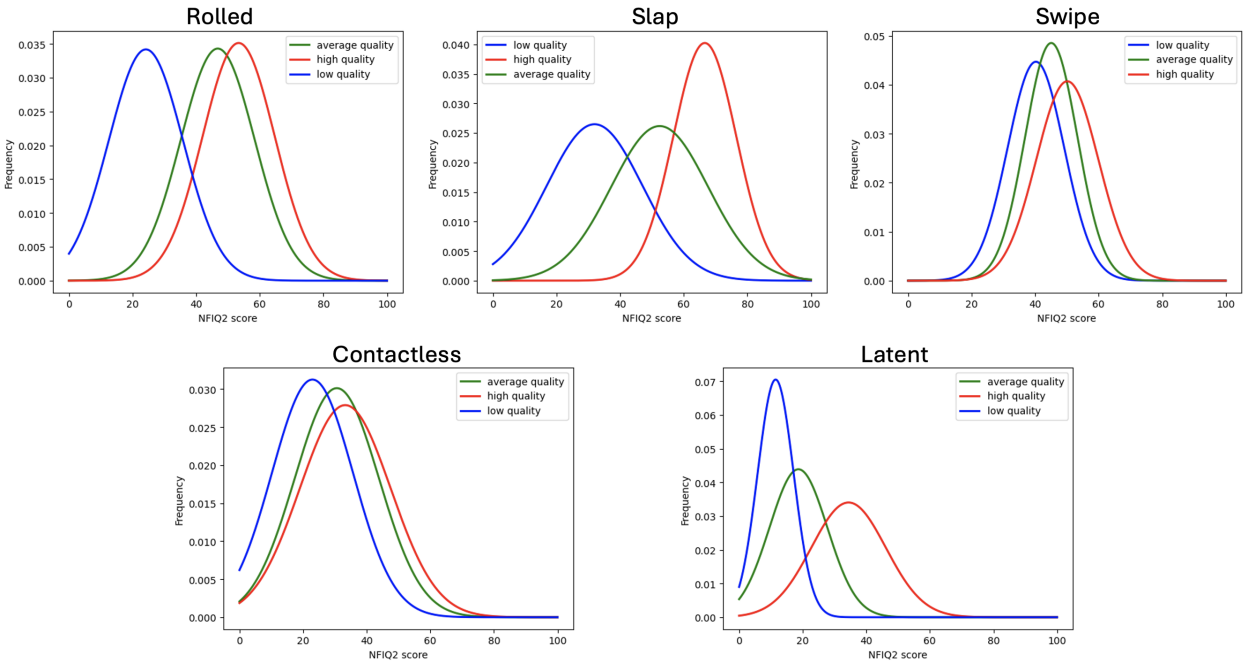


Figure 7.8 NFIQ 2.0 score distributions for fingerprints generated by GenPrint across five different fingerprint acquisition types.

7.4.3 Zero-shot Fingerprint Style Generation

To validate the quality of zero-shot fingerprint style generation, we performed one last experiment using t-SNE visualizations where we embed 100 example synthetic and real images from 6 different acquisition device domains from an unseen dataset which was not included in the training dataset for GenPrint. These images come from the recently released LFIW dataset [148]. Again, we observe very close similarity to corresponding real and synthetic images of the same acquisition devices, demonstrating GenPrint's adaptability toward zero-shot style generation from novel acquisition devices.

7.4.4 Utility for Training Fingerprint Recognition Models

One of the most important criteria for the quality of synthetic fingerprint generators is their utility for training fingerprint recognition models. We evaluate the utility of GenPrint both when training on only synthetically generated images and when augmenting a set of real fingerprint images with additional synthetic data. We compare with several previous synthetic fingerprint generators as baselines including SFinGe, PrintsGAN, and FPGAN-Control.

For the first experiment, we generate synthetic databases of 35,000 identities and 15 impressions per identity using each synthetic generation method. Some example images from each method are shown in Figure 7.1. We then train ResNet50 [106] recognition models using an ArcFace loss function on incremental subsets of each database using increments from 1,600 identities (the size of the real N2N fingerprint database) to 35,000 identities and plot the performance of the trained models on various evaluation datasets (see Figure 7.9). The evaluation datasets used are summarized in Table 7.3 and include fingerprint impressions of diverse acquisition devices including rolled, slap, contactless, and latent fingerprint types. We also summarize the TAR at an FAR of 0.1% for training on 35,000 identities from each method in Table 7.5. From Figure 7.9, we can clearly see that the performance of the recognition model trained on GenPrint images performs far better than any of the baseline synthetic methods and even surpasses the performance of training on the real N2N fingerprint dataset as the number of synthetic identities is increased.

For the second experiment, we compared the utility of GenPrint to the next best performing

synthetic method FPGAN-Control in augmenting an existing set of real fingerprint data for training on a combination of real and synthetic. Starting from the initial set of 1,600 real finger identities from N2N, we add increasing amounts of synthetic identities and again plot the performance of the trained ResNet50 models as the number of identities is increased. The results in Figure 7.10 show that both synthetic methods improve the performance when used for augmentation, but the improvement from GenPrint images is far superior.

The previous experiment showcased improvement of augmenting a limited set of real fingerprint data of only 1,600 unique finger identities, but naturally a question arises as to whether synthetic data augmentation is still helpful if the number of unique, real fingerprint identities in the training set is already large (e.g., 35,000). To investigate this question given that the number of identities is already large, rather than include additional synthetic identities, we instead take the existing real identities and use GenPrint to synthesize additional impressions in a more diverse range of acquisition devices. For this experiment, we use 35,000 unique fingerprint identities from the Michigan State Police (MSP) longitudinal fingerprint dataset [262] which has about 12 impressions per identity and augment each finger identity with an additonal 15 synthetic impressions of various acquisition devices. The result of augmenting MSP with GenPrint impressions is shown in Figure 7.10. As the number of identities increases, the plots show that GenPrint does indeed improve the performance significantly by augmenting the diversity of the already existing fingerprint images. This improvement is particularly evident when the test datasets contain sensor characteristics not included in the original MSP dataset but which GenPrint is able to synthesize (e.g., contactless and latent fingerprints). For reference, the TAR at FAR=0.1% using 35,000 training identities is given in Table 7.5.

In both of the previous experiments, we trained only ResNet50 models for the comparison. Thus, we now study the impact of additional model architectures and examine whether similar trends arise. In particular, we train two additional model architectures on both GenPrint and FPGAN-Control datasets of 35,000 identities and 15 impressions per identity. These include a ResNet18 model and a vision transformer (ViT) [64] with a patch size of 16 and 12 layers. As

Table 7.3 Test Datasets.

Test Dataset	Acquisition Types	Sensor Types	No. Images (Fingers)
PolyU Contactless 2D to Contact-based 2D [143]	Slap, Contactless	Digital Persona, smartphone	1920 (336)
NIST SD 4 [248]	Rolled	Ink on paper	4000 (2,000)
NIST SD 302 [78]	Slap, Contactless, Rolled,	CrossMatch, Eikon, GreenBit, ANDI, S120, MorphoWave, DactyScan, LIVETOUCH, Futronic, RaspiReader	2548 (400)
NIST SD 27 [83]	Rolled, Latent	Ink on paper, crime scene	1032 (258)

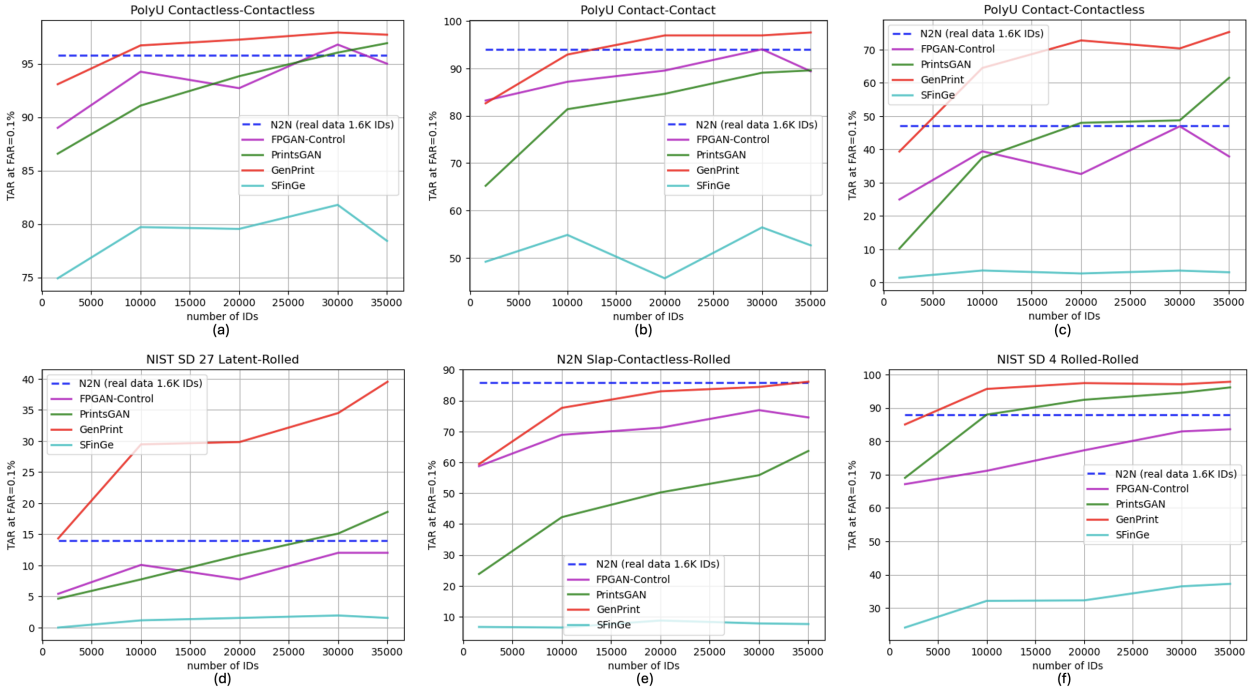


Figure 7.9 Authentication accuracy (TAR at FAR=0.1%) of ResNet50 trained on synthetic data from various fingerprint generation methods including the proposed GenPrint.

shown in Table 7.6, the same relative performance gap between training on GenPrint vs. FPGAN-Control images across each model architecture is consistent with our more extensive experiments using ResNet50.

Table 7.4 Authentication accuracy (TAR at FAR=0.1%) of ResNet50 trained on synthetic data from various fingerprint generators including the proposed GenPrint. A ResNet50 model trained on N2N, a real dataset, is included as a baseline.

Training Data	No. IDs	No. imgs/ID	N2N slap-rolled-contactless	NIST SD4 rolled-rolled	PolyU contact-contact	PolyU contactless-contactless	PolyU contact-contactless	NIST SD27 latent-rolled
N2N [78] (real dataset)	1,600	12	85.73	87.9	94	95.79	47.08	13.95
SFinGe [33]	35,000	15	7.62	37.25	52.63	78.42	3.07	1.55
FPGAN-Control [214]	35,000	15	74.52	83.6	89.38	95	37.86	12.02
PrintsGAN [71]	35,000	15	63.66	96.15	89.58	96.92	61.51	18.6
GenPrint	35,000	15	**86.08**	**97.85**	**97.58**	**97.71**	**75.26**	**39.53**

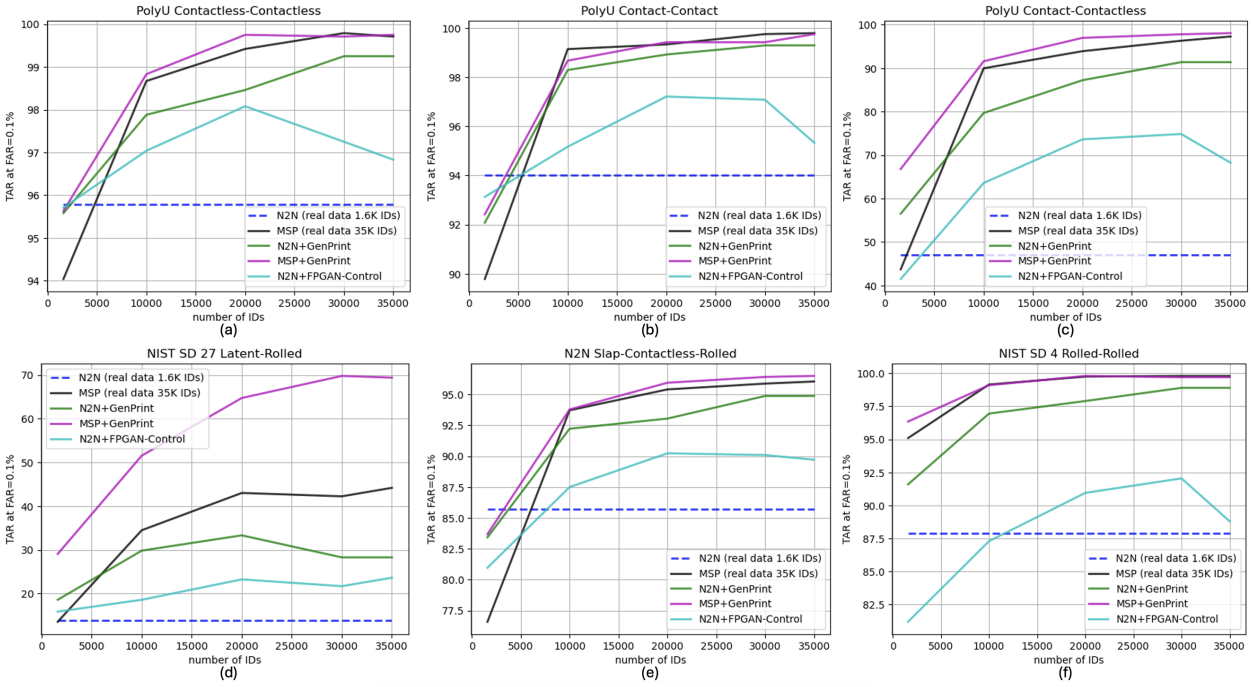


Figure 7.10 Authentication accuracy (TAR at FAR=0.1%) of ResNet50 trained on a combination of real and synthetic data from FPGAN-Control and the proposed GenPrint evaluated on six different test scenarios.

Table 7.5 Authentication accuracy (TAR at FAR=0.1%) of ResNet50 trained on a combination of real and synthetic data from FPGAN-Control and the proposed GenPrint evaluated on six different test scenarios.

Training Data	No. IDs	No. imgs/ID	N2N slap-rolled-contactless	NIST SD4 rolled-rolled	PolyU contact-contact	PolyU contactless-contactless	PolyU contact-contactless	NIST SD27 latent-rolled
N2N [78] (real dataset)	1,600	12	85.73	87.9	94	95.79	47.08	13.95
N2N [78] + FPGAN [214]	35,000	15	89.71	88.8	95.33	96.83	68.25	23.64
N2N [78] + GenPrint	35,000	13.5	94.69	98.9	99.54	99.17	90.9	46.51
MSP [262] (real dataset)	35,000	12	96.04	**99.80**	**99.79**	99.71	97.29	62.02
MSP [262] + GenPrint	35,000	27	**96.49**	99.70	99.75	**99.75**	**98.07**	**69.38**

Table 7.6 Training on 35K IDs, 15 impressions with different model architectures and evaluated on six different test scenarios. Results reported as TAR @ FAR=0.1%.

Model	Training Dataset	N2N slap-rolled-contactless	NIST SD4 rolled-rolled	PolyU contact-contact	PolyU contactless-contactless	PolyU contact-contactless	NIST SD27 latent-rolled
ResNet18	FPGAN [214]	61.92	70.6	80.79	87.83	26.70	8.14
ResNet18	GenPrint	**71.84**	**91.75**	**90.25**	**93.04**	**42.45**	**23.64**
ResNet50	FPGAN [214]	74.52	83.60	89.38	95.00	37.86	12.02
ResNet50	GenPrint	**86.08**	**97.85**	**97.58**	**97.71**	**75.26**	**39.53**
ViT	FPGAN [214]	45.47	70.40	45.92	83.33	6.28	5.43
ViT	GenPrint	**79.81**	**96.80**	**95.75**	**96.71**	**61.25**	**30.62**

7.4.5 Utility for Evaluating Fingerprint Recognition Models

In addition to being useful for training, synthetic fingerprints can also help with large-scale evaluation of fingerprint recognition algorithms, where collecting a dataset of potentially millions of unique real fingers can be prohibitively expensive. To demonstrate the feasibility of GenPrint images to be used for such purposes, we generated a large database of 64,000 unique rolled fingerprints to compare with a database of 64,000 real rolled fingerprint identities from the MSP dataset as a background gallery for latent to rolled fingerprint search using latent probes and corresponding mates from the NIST SD 27 latent dataset. Ideally, the search performance should

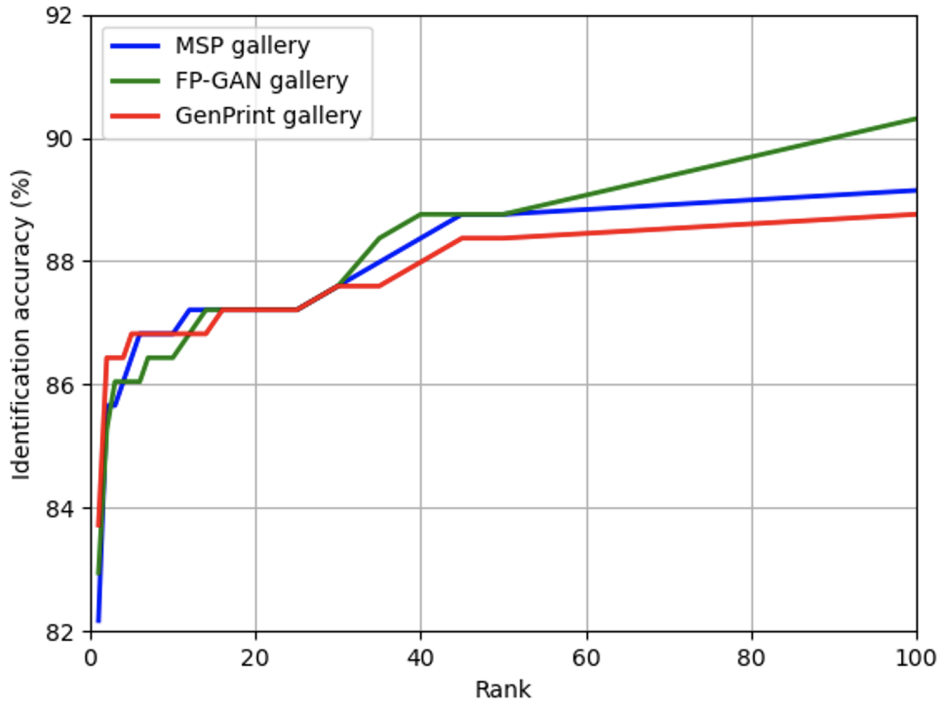


Figure 7.11 Search results using probes from NIST SD 27 and 64,000 identity background from GenPrint compared to the real MSP dataset and FPGAN-Control backgrounds.

be similar when using the real fingerprint background images and GenPrint fingerprint background images. We repeated the experiment using a database of 64,000 unique identities from FPGAN-Control as a baseline. The results on the three different gallery backgrounds are given in Figure 7.11, which shows better overlap in the search accuracies between GenPrint background gallery and the real fingerprint gallery compared to the overlap between FPGAN-Control and the real dataset, indicating that GenPrint images make a more suitable replacement for real images for large-scale search evaluations than the baseline FPGAN-Control method. In particular, the rank-1 accuracy on the real background dataset is 82.17%, whereas it was 82.95% and 83.72% for GenPrint and FPGAN-Control, respectively.

7.4.6 Biometric Capacity

Ideally, every synthetic finger identity should be unique, but the probability of encountering “duplicate" identities, those which have a high similarity to each other, increases as the size of the dataset grows, which is true even for real fingerprint datasets. Nonetheless, the possible number of unique finger identities that a model can generate, referred to as the biometric capacity [18], is an

important factor for comparison among synthetic biometric generators. Unfortunately, accurately measuring the biometric capacity is a difficult and open question, and empirically computing the similarity between all generated identities scales in complexity of O(n^2), making it computationally demanding for anything above 100,000 identities.

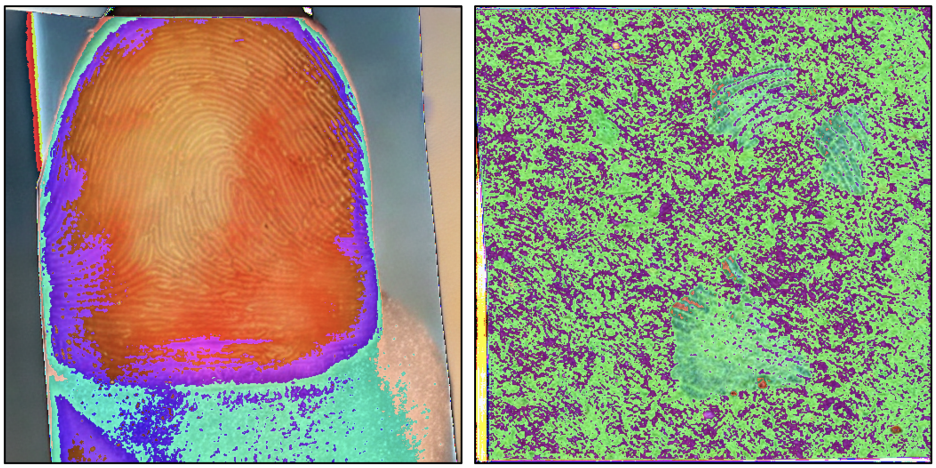
Recently, Bodetti et al. [18] proposed a geometrical model of capacity by embedding face images into a hyperspherical representation space and used a specified FAR to estimate the ratio of the overall embedding space volume and the intra-class separation. However, this approach only aims to estimate an upper bound on the capacity, and when we applied the code to our generated images and several other baselines, we received capacity estimates on the order of 10^{32}. To obtain more practical insights, we used a pretrained AFR-Net fingerprint matcher to compute the percentage of "duplicate" identities as the number of generated identities increases for both PrintsGAN and GenPrint. We obtained duplicate identities by computing all possible imposter score comparisons between the generated identities and determined how many of the pairs produced similarity scores to each other which fell above the genuine match threshold of 0.35 computed on the real NIST SD4 dataset for a FAR of 0.01%. As shown in Table 7.7, the number of duplicate identities is increasing at a large rate for PrintsGAN as the number of generated identities increases, whereas GenPrint closely follows the trend on the real MSP fingerprint dataset as the number of identities approaches 100,000. Interestingly, PrintsGAN and GenPrint were both trained on fingerprint databases of a similar number of identities (38,291 for PrintsGAN and 37,351 for GenPrint), the difference being that GenPrint is based-off diffusion models which are believed to better capture the full data distribution compared to GANs [57].

7.4.7 Identity Leakage

Besides the capacity of the biometric generator, a privacy preserving model should also not leak sensitive information from the dataset on which it was trained. In other words, the generated finger identities from GenPrint should not have high similarity with any of the finger identities in the training set. We aimed to measure the potential identity leakage of GenPrint by generating 35,000 unique synthetic fingerprint identities and computed similarity scores to each of the 37,351 real

Table 7.7 Percentage of duplicate identities generated by PrintsGAN and GenPrint as the number of generated identities increases from 20,000 to 100,000. A duplicate identity is counted whenever an imposter score between any of the generated identities is above a genuine match threshold of 0.35, which was computed on the real NIST SD4 dataset using a pretrained AFR-Net fingerprint recognition model.

Number of IDs	MSP (real data)	GenPrint	PrintsGAN [71]
20,000	0.060%	0.070%	4.255%
40,000	0.078%	0.190%	7.408%
60,000	0.103%	0.305%	9.885%
80,000	0.154%	0.461%	12.23%
100,000	0.170%	0.535%	14.43%



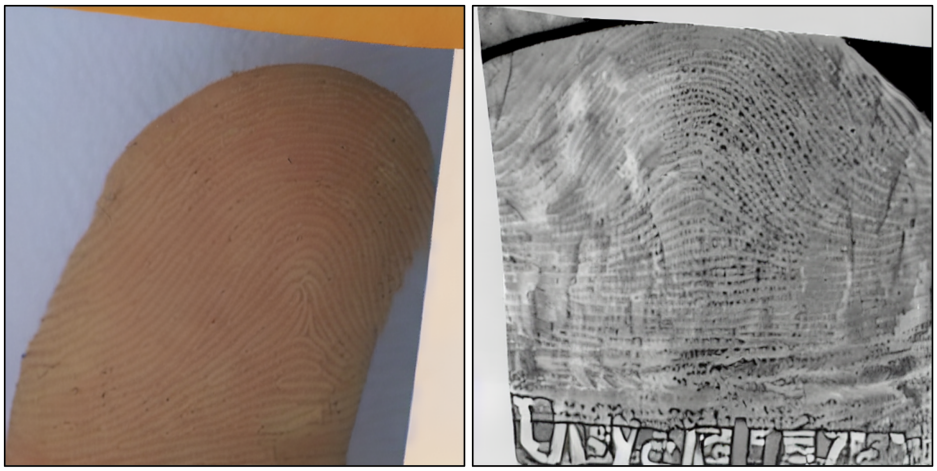
(a) (b)

Figure 7.12 Example failure cases generated by GenPrint exhibiting noise and color artifacts.

training finger identities in our training dataset using a pretrained AFR-Net fingerprint recognition model [97]. Out of these 35,000 synthetic identities, only 10 (0.03%) had a similarity score with any training identity above 0.231, the genuine match threshold computed on FVC 2002 DB1A at FAR=0.01%. Furthermore, even out of those ten similarity scores that fell above the threshold, the maximum similarity score obtained was just 0.297, only slightly above the threshold.

7.4.8 Failures and Limitations

On occasion, the outputs generated from GenPrint can exhibit some noise and other color artifacts. Some of these failure cases are visualized in Figure 7.12. For the image in subfigure (a), the prompt was for a low quality contactless fingerprint from a smartphone camera, and the prompt



(a) (b)

Figure 7.13 Example images with mixed text prompts. The image in (a) was created with a prompt containing the acquisition type of “latent" and sensor type of “smartphone", whereas (b) was prompted with acquisition type “contactless" and sensor type “crime scene".

used in (b) was for a low quality latent fingerprint from a crime scene. Empirically, we found that the probability of such artifacts occurring is higher when the quality of the fingerprint is prompted as low. Another potential area for unexpected outputs is in mixing acquisition and sensor types that may not be realistic. For example, the image produced in subfigure (a) of Figure 7.13 was prompted with the acquisition type of “latent" and sensor type of “smartphone", whereas subfigure (b) was prompted with acquisition type “contactless" and sensor type “crime scene". These mixed prompts produce fingerprint images that resemble characteristics of both contactless and latent fingerprints but are not quite realistic. Nonetheless, even though mixing various acquisition and sensor types may produce unrealistic fingerprint images, the results may still prove to be a useful data augmentation tool for training fingerprint recognition models. In fact, all experiments conducted in this paper are without editing or removing any images generated by GenPrint.

One of the most significant limitations of GenPrint and DDPM models in general is the computational efficiency, both in terms of training and inference time and memory footprint. Training efficiency of GenPrint was partially mitigated by utilizing LoRA weights for training; however, the inference speed for GenPrint is still about 1.13 seconds per image (512×512 resolution) using an Nvidia A100 GPU and an AMD EPYC 7543 32-Core Processor - which is much slower compared to

GANs (e.g., 7.41 ms per image for FPGAN-Control on the same hardware). For offline generation of synthetic datasets, the latency and RAM usage are both just a nuisance; however, they prevent the model from being useful for online generation of synthetic data during training of recognition models.

7.5 Conclusion

By employing latent diffusion models with multimodal conditions, GenPrint offers a versatile framework capable of generating diverse fingerprint images while preserving identity and providing explainable control over various appearance factors. Unlike previous approaches, GenPrint is not constrained by the characteristics of the training dataset alone, allowing for the generation of novel sensor and style attributes during inference without the need for additional fine-tuning. The experimental results showcase the efficacy of GenPrint in terms of identity preservation and narrowing the gap between synthetic and real domains. Moreover, the universality of GenPrint-generated images improves model training by augmenting the diversity of existing fingerprint datasets, thus enhancing the performance and generalization of fingerprint recognition systems. The same or similar model architecture can also be applied to other areas of biometrics (e.g., face, palmprint, iris, etc.) which we are currently undertaking.

7.6 Acknowledgment

Parts of this research were supported by a grant from the Department of Homeland Security via The Criminal Investigations and Network Analysis Center (CINA) at George Mason University.

CHAPTER 8

SUMMARY

8.1 Contributions

In this dissertation, we have presented several methods to improve the generalization of fingerprint embeddings to various challenging and unconstrained scenarios. These specific scenarios and contributions are enumerated below:

- **Sensor and material agnostic fingerprint PAD:**
  - A robust PAD solution with improved cross-material and cross-sensor generalization performance.
  - The proposed solution using style transfer and adversarial representational learning can be built on top of any CNN-based fingerprint PAD solution for cross-sensor and cross-material PA generalization.
  - Experimental evaluation of the proposed method on publicly available datasets LivDet 2015, LivDet 2017, MSU-FPAD, and GCT3. The approach is shown to improve the cross-sensor (cross-sensor and cross-material) generalization performance from a TDR of 88.36% (78.76%) to a TDR of 93.03% (88.49%) at a FDR of 0.2%.
- **Contact to contactless compatible fingerprint recognition:**
  - An end-to-end system, called C2CL, for contact-contactless fingerprint matching. C2CL is comprised of preprocessing (segmentation, enhancement, scaling, and deformation correction), feature extraction (minutiae and texture representations), and matching modules aimed at reducing the domain gap between contact and corresponding contactless fingerprint images.
  - Our preprocessing is generalizable as it was shown to also benefit the Verifinger 12.0 commercial fingerprint SDK.
  - A contact-contactless adaptation of DeepPrint [69] for representation extraction. Our representation is generalizable across multiple datasets and contactless capture devices.
  - SOTA cross-matching verification and large-scale identification accuracy using C2CL

on both publicly available contact-contactless matching datasets as well as on a completely sequestered dataset collected at Zhejiang University, China. Our evaluation includes the most diverse set of contactless fingerprint acquisition devices, yet we employ just a single trained model for evaluation.

  - A smartphone contactless fingerprint capture app that was developed in-house for improved throughput and user-convenience. This app will be made available to the public to promote further research in this area.
  - A new dataset of $9,888$ 2D contactless and corresponding contact-based fingerprint images from 206 subjects (2 thumbs and 2 index fingers per subject), made publicly available to advance much needed research in this area.

- **Multimodel embeddings for improved sensor-interoperability of fingerprint recognition:**
  - Analysis of various attention-based architectures for fingerprint recognition.
  - Novel architecture for fingerprint recognition, AFR-Net, which incorporates attention layers into the ResNet architecture.
  - State-of-the-art (SOTA) fingerprint recognition performance (authentication and identification) across several diverse benchmark datasets, including intra-sensor, cross-sensor, contact to contactless, and latent to rolled fingerprint matching.
  - Novel use of local embeddings extracted from intermediate feature maps to both improve the recognition accuracy and explainability of the model.
  - Ablation analysis demonstrating the importance of each aspect of our model, including choice of loss function, training dataset size, use of spatial alignment module, use of both classification heads, and use of local embeddings to refine the global embeddings.
- **Improved latent fingerprint recognition via fusion of global and local embeddings:**
  - Design of an end-to-end latent fingerprint recognition pipeline using deep learning

The project repository for the smartphone contactless fingerprint capture app is available at https://github.com/ronny3050/FingerPhotos.
The dataset application is available at https://person.zju.edu.cn/en/eryunliu.

methods, including algorithms for segmentation, enhancement, minutiae extraction, and a fusion of global and local embeddings.

  - State-of-the-art (SOTA) latent to rolled/plain fingerprint search across multiple datasets, including NIST SD 27 [83], NIST SD 302 Latents (N2N Latents) [78], MSP Latent [262], and MOLF datasets [212].
  - Faster search speed (low latency) due to our multi-stage search scheme, while maintaining SOTA recognition accuracy for both closed-set and open-set identification.
  - Generalization of representation (embedding) from LFR-Net is shown via SOTA authentication performance across several rolled (NIST SD 14 [246]), plain (NIST SD 302 [78]), and contact to contactless fingerprint matching datasets (PolyU Contactless 2D to Contact-based 2D [145] and ZJU Finger Photo and Touch-based [93]) using the same network, a step toward a universal fingerprint recognition system.

- **Synthetic fingerprint spoof images for improved fingerprint PAD:**
  - A highly realistic plain print synthetic fingerprint generator capable of generating multiple impressions per finger.
  - The first, to the best of our knowledge, synthetic fingerprint PA generator which is capable of producing synthetic representations of both bona fide and PA impressions of the same finger. This opens the door to joint optimization of fingerprint PAD and recognition algorithms.
  - Quantitative and qualitative analysis to verify the quality of our generated bona fide and PA fingerprints.
  - Experiments showcasing improved fingerprint PAD on both seen and unseen PA material types when augmenting existing fingerprint datasets with our synthetic bona fide and PA fingerprints.
  - We release our code and a database of SpoofGAN images to encourage further research in this area https://github.com/groszste/SpoofGAN.

- **Universal Fingerprint Generation:**

- A controllable latent diffusion model, GenPrint, using text and image conditions for highly realistic and diverse synthetic fingerprint generation.
- GenPrint is capable of generating fingerprints of any acquisition type, sensor, fingerprint class, and quality, including fingerprint styles not seen during training without any additional fine-tuning (e.g., zero-shot fingerprint style generation).
- The generation process is controllable (both in appearance and identity preservation) and explainable with humanly interpretable text prompts.
- The utility of GenPrint synthetic images is validated through experiments showcasing improved recognition performance of models trained on GenPrint images compared to real datasets and other benchmark fingerprint generation methods.
- We also demonstrate the utility of GenPrint images for evaluating fingerprint recognition systems by replacing real data for large-scale identification experiments.
- Open-sourcing a dataset of 100K synthetic finger identities with 15 impressions of various acquisition devices to the research community.

8.2 Suggestions for Future Work

The following are some of the possible future directions within the scope of improving the generalization of fingerprint embeddings:

- **Infant fingerprint recognition:** Recognition of infant fingerprints has shown some initial promise on recognizing infants enrolled between the ages of 2-3 months [70]; however, the performance still significantly lags that of adult fingerprint recognition. Methods presented in this dissertation, particularly style transfer, fusion of multiple deep networks, and fusion of local and global features, may be applicable to this scenario; however, further advanced methods specifically tailored to infant fingerprints is an area of important future research.
- **Mobile-based fingerprint recognition:** With extending fingerprint recognition to applications involving mobile devices, there is a need for computationally efficient fingerprint recognition algorithms which can be embedded into the mobile device. This would facilitate applications such as integrated image quality assessment, deduplication, and recognition in

real-time, as well as preserve the privacy of users by alleviating the need to send images and/or templates over the network.

8.3 List of Publications

A list of publications during the course of my PhD program related to the topics of this thesis:

Journal Articles

- **S. A. Grosz**, and A. K. Jain, “Universal Fingerprint Generation: Controllable Diffusion Model with Multimodal Conditions", under review in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024. Patent pending.
- **S. A. Grosz**, A. Godbole, and A. K. Jain, “Mobile Contactless Palmprint Recognition: Use of Multiscale, Multimodel Embeddings", under review in IEEE Transactions on Information Forensics and Security, 2024.
- **S. A. Grosz**, and A. K. Jain, “Latent Fingerprint Recognition: Fusion of Local and Global Embeddings", IEEE Trans. Information Forensics and Security, vol. 18, pp. 5691-5705, 2023. Technology licensed to Thales. Patent pending.
- **S. A. Grosz**, and A. K. Jain, “AFR-Net: Attention-Driven Fingerprint Recognition Network", IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 6, pp. 30-42, 2023.
- K. P. Wijewardena, **S. A. Grosz**, and A. K. Jain, “Fingerprint Template Invertibility: Minutiae vs. Deep Templates", IEEE Trans. Information Forensics and Security, vol. 18, pp. 744-757, 2023.
- **S. A. Grosz**, and A. K. Jain, “SpoofGAN: Synthetic Fingerprint Spoof Images", IEEE Trans. Information Forensics and Security, vol. 18, pp. 730-743, 2023.
- J. J. Engelsma, **S. A. Grosz**, and A. K. Jain, “PrintsGAN: Synthetic Fingerprint Generator", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 5, pp. 6111-6124, 2023. Technology licensed to NEC.
- **S. A. Grosz**, J. J. Engelsma and A. K. Jain, “C2CL: Contact to Contactless Fingerprint Matching", IEEE Trans. Information Forensics and Security, vol. 17, pp. 196-210, 2022.

Conference Proceedings and Technical Reports

- A. Godbole, **S. A. Grosz**, and A. K. Jain, “Contactless Palmprint Recognition for Children", International Conference of the Biometrics Special Interest Group (BIOSIG), Sept. 2023.
- **S. A. Grosz**, K. P. Wijewardena, and A. K. Jain, “ViT Unified: Joint Fingerprint Recognition and Presentation Attack Detection", IEEE International Joint Conference on Biometrics, Ljubljana, Sept. 2023.
- **S. A. Grosz**, J. J. Engelsma and A. K. Jain, “White-Box Evaluation of Fingerprint Recognition Systems", ArXiv, 2021, arXiv:2008.00128.
- **S. A. Grosz**, J. J. Engelsma, N. G. Paulter and A. K. Jain, “White-Box Evaluation of Fingerprint Matchers: Robustness to Minutiae Perturbations”, IEEE International Joint Conference on Biometrics, Houston, TX, Sept. 2020
- **S. A. Grosz**, T. Chugh and A. K. Jain, “Fingerprint Presentation Attack Detection: A Sensor and Material Agnostic Approach", IEEE International Joint Conference on Biometrics, Houston, TX, Sept. 2020.

BIBLIOGRAPHY

[1] Aadhaar - Unique Identification Authority of India. https://uidai.gov.in/. Accessed: January 16, 2024.

[2] ELFT-1.X Results. https://pages.nist.gov/elft/elft_1_x/results/. Accessed: May 3, 2024.

[3] M. B. Alejo. Unconstrained ear recognition using transformers. *Jordanian Journal of Computers and Information Technology*, 7(4), 2021.

[4] F. Alonso-Fernandez, R. N. Veldhuis, A. M. Bazen, J. Fiérrez-Aguilar, and J. Ortega-Garcia. Sensor interoperability and fusion in fingerprint verification: A case study using minutiae- and ridge-based matchers. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6. IEEE, 2006.

[5] H. AlShehri, M. Hussain, H. AboAlSamh, and M. AlZuair. A large-scale study of fingerprint matching systems for sensor interoperability problem. *Sensors*, 18(4):1008, 2018.

[6] A. H. Ansari. Generation and storage of large synthetic fingerprint database. *ME Thesis, Jul*, 2011.

[7] S. R. Arashloo, J. Kittler, and W. Christmas. An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol. *IEEE Access*, 5:13868–13882, 2017.

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

[9] M. Attia, M. H. Attia, J. Iskander, K. Saleh, D. Nahavandi, A. Abobakr, M. Hossny, and S. Nahavandi. Fingerprint synthesis via latent space representation. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1855–1861. IEEE, 2019.

[10] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[11] K. Bahmani, R. Plesh, P. Johnson, S. Schuckers, and T. Swyka. High fidelity fingerprint generation: Quality, uniqueness, and privacy. *arXiv preprint arXiv:2105.10403*, 2021.

[12] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni. Fake fingerprint detection by odor analysis. In *Proc. ICB*, pages 265–272. Springer, 2006.

[13] A. M. Bazen and S. H. Gerez. Fingerprint matching by thin-plate spline modelling of elastic deformations. *Pattern Recognition*, 36(8):1859–1867, 2003.

[14] N. Beheshti and L. Johnsson. Squeeze u-net: A memory and energy efficient image segmentation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern*

recognition workshops, pages 364–365, 2020.

[15] S. M. Bellovin, P. K. Dutta, and N. Reitinger. Privacy and synthetic datasets. *Stan. Tech. L. Rev.*, 22:1, 2019.

[16] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[17] P. Birajadar, M. Haria, P. Kulkarni, S. Gupta, P. Joshi, B. Singh, and V. Gadre. Towards smartphone-based touchless fingerprint recognition. *Sādhanā*, 44(7):1–15, 2019.

[18] V. N. Boddeti, G. Sreekumar, and A. Ross. On the biometric capacity of generative face models. In *2023 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2023.

[19] P. Bontrager, A. Roy, J. Togelius, N. Memon, and A. Ross. Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution. In *IEEE 9th International Conference on Biometrics Theory, Applications and Systems*, pages 1–9. IEEE, 2018.

[20] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[21] R. Bouzaglo and Y. Keller. Synthesis and reconstruction of fingerprints using generative adversarial networks. *arXiv preprint arXiv:2201.06164*, 2022.

[22] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[23] K. Cao and A. Jain. Fingerprint synthesis: Evaluating fingerprint search at scale. In *International Conference on Biometrics*, 2018.

[24] K. Cao and A. K. Jain. Latent orientation field estimation via convolutional neural network. In *2015 International Conference on Biometrics (ICB)*, pages 349–356. IEEE, 2015.

[25] K. Cao and A. K. Jain. Hacking mobile phones using 2D Printed Fingerprints. MSU Tech. report, MSU-CSE-16-2 https://www.youtube.com/watch?v=fZJI_BrMZXU, 2016.

[26] K. Cao and A. K. Jain. Fingerprint indexing and matching: An integrated approach. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 437–445. IEEE, 2017.

[27] K. Cao and A. K. Jain. Fingerprint Indexing and Matching: An Integrated Approach. In *2017 IEEE International Joint Conference on Biometrics*, pages 437–445. IEEE, 2017.

[28] K. Cao and A. K. Jain. Automated latent fingerprint recognition. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, 41(4):788–800, 2018.

[29] K. Cao and A. K. Jain. Latent fingerprint recognition: role of texture template. In *2018 IEEE 9th international conference on biometrics theory, applications and systems (BTAS)*, pages 1–9. IEEE, 2018.

[30] K. Cao, E. Liu, and A. K. Jain. Segmentation and enhancement of latent fingerprints: A coarse to fine ridgestructure dictionary. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1847–1859, 2014.

[31] K. Cao, D.-L. Nguyen, C. Tymoszek, and A. K. Jain. End-to-end latent fingerprint search. *IEEE Transactions on Information Forensics and Security*, 15:880–894, 2019.

[32] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt. *arXiv preprint arXiv:2303.04226*, 2023.

[33] R. Cappelli, A. Erol, D. Maio, and D. Maltoni. Synthetic fingerprint-image generation. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, pages 471–474. IEEE, 2000.

[34] R. Cappelli, M. Ferrara, and D. Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2128–2141, 2010.

[35] R. Cappelli, D. Maio, and D. Maltoni. Modelling plastic distortion in fingerprint images. In *International Conference on Advances in Pattern Recognition*, pages 371–378. Springer, 2001.

[36] R. Cappelli, D. Maio, and D. Maltoni. Synthetic fingerprint-database generation. In *Object recognition supported by user interaction for service robots*, volume 3, pages 744–747. IEEE, 2002.

[37] R. Cappelli, D. Maio, and D. Maltoni. Semi-automatic enhancement of very low quality fingerprints. In *2009 Proceedings of 6th International Symposium on Image and Signal Processing and Analysis*, pages 678–683. IEEE, 2009.

[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[39] R. Casula, M. Micheletto, G. Orrù, R. Delussu, S. Concas, A. Panzino, and G. L. Marcialis. Livdet 2021 fingerprint liveness detection competition-into the unknown. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–6. IEEE, 2021.

[40] S. Chikkerur, A. N. Cartwright, and V. Govindaraju. Fingerprint enhancement using stft analysis. *Pattern Recognition*, 40(1):198–211, 2007.

[41] T. Chugh, K. Cao, and A. K. Jain. Fingerprint spoof buster: Use of minutiae-centered patches. *IEEE Transactions on Information Forensics and Security*, 13(9):2190–2202, September 2018.

[42] T. Chugh and A. K. Jain. Fingerprint Presentation Attack Detection: Generalization and Efficiency. *IEEE International Conference on Biometrics (ICB)*, pages 1–8, 2019.

[43] T. Chugh and A. K. Jain. OCT Fingerprints: Resilience to Presentation Attacks. *arXiv preprint arXiv:1908.00102*, 2019.

[44] T. Chugh and A. K. Jain. Fingerprint spoof detector generalization. *IEEE Transactions on Information Forensics and Security*, 16:42–55, 2020.

[45] Cisco. Cisco report: 81% of all smartphones have biometrics enabled. https://www.biometricupdate.com/202211/cisco-report-81-percent-of-all-smartphones-have-biometrics-enabled, 2022. Accessed on: January 16, 2024.

[46] G. Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[47] A. Dabouei, H. Kazemi, S. M. Iranmanesh, J. Dawson, and N. M. Nasrabadi. Fingerprint distortion rectification using deep convolutional neural networks. In *2018 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2018.

[48] A. Dabouei, H. Kazemi, S. M. Iranmanesh, J. Dawson, N. M. Nasrabadi, et al. Id preserving generative adversarial network for partial latent fingerprint reconstruction. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–10. IEEE, 2018.

[49] A. Dabouei, S. Soleymani, J. Dawson, and N. M. Nasrabadi. Deep contactless fingerprint unwarping. In *International Conference on Biometrics*, pages 1–8. IEEE, 2019.

[50] H. M. Daluz. *Fundamentals of fingerprint analysis*. CRC Press, 2018.

[51] L. N. Darlow and B. Rosman. Fingerprint minutiae extraction using deep learning. In *IEEE International Joint Conference on Biometrics*, pages 22–30. IEEE, 2017.

[52] A. K. Datta, H. C. Lee, R. Ramotowski, and R. Gaensslen. *Advances in fingerprint technology*. CRC press, 2001.

[53] D. Deb, T. Chugh, J. Engelsma, K. Cao, N. Nain, J. Kendall, and A. K. Jain. Matching fingerphotos to slap fingerprint images. *arXiv preprint arXiv:1804.08122*, 2018.

[54] P. Delgado-Santos, R. Tolosana, R. Guest, F. Deravi, and R. Vera-Rodriguez. Exploring transformers for behavioural biometrics: A case study in gait recognition. *arXiv preprint arXiv:2206.01441*, 2022.

[55] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pages 837–845, 1988.

[56] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[57] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[58] Y. Ding and A. Ross. An ensemble of one-class SVMs for fingerprint spoof detection across different fabrication materials. In *Proc. IEEE WIFS*, pages 1–6, 2016.

[59] Y. Ding and A. Ross. An ensemble of one-class svms for fingerprint spoof detection across different fabrication materials. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2016.

[60] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[61] X. Dong and J. Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European conference on computer vision*, pages 459–474, 2018.

[62] B. Dorizzi, R. Cappelli, M. Ferrara, D. Maio, D. Maltoni, N. Houmani, S. Garcia-Salicetti, and A. Mayoue. Fingerprint and on-line signature verification competitions at ICB 2009. In *International Conference on Biometrics*, pages 725–732. Springer, 2009.

[63] B. Dorizzi, R. Cappelli, M. Ferrara, D. Maio, D. Maltoni, N. Houmani, S. Garcia-Salicetti, and A. Mayoue. Fingerprint and on-line signature verification competitions at icb 2009. In *International Conference on Biometrics*, pages 725–732. Springer, 2009.

[64] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[65] H. Edwards and A. Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

[66] J. J. Engelsma, S. S. Arora, A. K. Jain, and N. G. Paulter. Universal 3D wearable Fingerprint Targets: Advancing Fingerprint Reader Evaluations. *IEEE Transactions on Information*

Forensics and Security, 13(6):1564–1578, 2018.

[67] J. J. Engelsma, K. Cao, and A. K. Jain. Raspireader: Open Source Fingerprint Reader. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[68] J. J. Engelsma, K. Cao, and A. K. Jain. Learning a fixed-length fingerprint representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[69] J. J. Engelsma, K. Cao, and A. K. Jain. Learning a fixed-length fingerprint representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[70] J. J. Engelsma, D. Deb, K. Cao, A. Bhatnagar, P. S. Sudhish, and A. K. Jain. Infant-id: Fingerprints for global good. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3543–3559, 2021.

[71] J. J. Engelsma, S. A. Grosz, and A. K. Jain. Printsgan: synthetic fingerprint generator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[72] J. J. Engelsma and A. K. Jain. Generalizing fingerprint spoof detector: Learning a one-class classifier. In *International Conference on Biometrics*, pages 1–8. IEEE, 2019.

[73] J. J. Engelsma and A. K. Jain. Generalizing fingerprint spoof detector: Learning a one-class classifier. In *2019 International Conference on Biometrics (ICB)*, pages 1–8, 2019.

[74] J. J. Engelsma, A. K. Jain, and V. N. Boddeti. Hers: Homomorphically encrypted representation search. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2022.

[75] L. Ericson and S. Shine. Evaluation of contactless versus contact fingerprint data phase 2 (version 1.1). Technical Report 249552, DOJ Office Justice Programs, I. ManTech Adv. Syst. Int., Fairmont, WV, 2015.

[76] M. A. I. Fahim and H. Y. Jung. A lightweight gan network for large scale fingerprint generation. *IEEE Access*, 8:92918–92928, 2020.

[77] J. Feng, J. Zhou, and A. K. Jain. Orientation field estimation for latent fingerprint enhancement. *IEEE transactions on pattern analysis and machine intelligence*, 35(4):925–940, 2012.

[78] G. P. Fiumara, P. A. Flanagan, J. D. Grantham, K. Ko, K. Marshall, M. Schwarz, E. Tabassi, B. Woodgate, and C. Boehnen. NIST Special Database 302: Nail to Nail Fingerprint Challenge. Technical Report NIST.TN.2007, National Institute of Standards and Technology, Gaithersburg, MD, 2019.

[79] R. Gajawada, A. Popli, T. Chugh, A. Namboodiri, and A. K. Jain. Universal Material Translator: Towards Spoof Fingerprint Generalization. In *IEEE International Conference*

on Biometrics (ICB), 2019.

[80] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[81] F. Galton. *Finger prints*. Macmillan and Company, 1892.

[82] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[83] M. D. Garris and M. D. Garris. *NIST special database 27: Fingerprint minutiae from latent and matching tenprint images*. US Department of Commerce, National Institute of Standards and Technology, 2000.

[84] L. Ghiani, A. Hadid, G. L. Marcialis, and F. Roli. Fingerprint liveness detection using Binarized Statistical Image Features. In *Proc. IEEE 6th Int. Conf. BTAS*, pages 1–6, 2013.

[85] L. Ghiani, G. L. Marcialis, and F. Roli. Fingerprint liveness detection by local phase quantization. In *Proc. 21st ICPR*, pages 537–540, 2012.

[86] L. Ghiani, D. Yambay, V. Mura, S. Tocco, G. L. Marcialis, F. Roli, and S. Schuckcrs. LivDet 2013 Fingerprint Liveness Detection Competition 2013. In *Proc. ICB*, pages 1–6. IEEE, 2013.

[87] R. C. Gonzales and P. Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[88] L. J. González-Soler, M. Gomez-Barrero, L. Chang, A. Pérez-Suárez, and C. Busch. Fingerprint Presentation Attack Detection Based on Local Features Encoding for Unknown Attacks. *arXiv preprint arXiv:1908.10163*, 2019.

[89] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680, 2014.

[90] Google. Introducing imagen: A large-scale image library and open-source tooling to enable computer vision research, May 2021.

[91] P. W. Greenwood and J. Petersiua. The criminal investigation process volume i: Summary and policy impucations. 1975.

[92] S. A. Grosz, T. Chugh, and A. K. Jain. Fingerprint presentation attack detection: A sensor and material agnostic approach. In *IEEE International Joint Conference on Biometrics*,

pages 1–10. IEEE, 2020.

[93] S. A. Grosz, J. J. Engelsma, E. Liu, and A. K. Jain. C2cl: Contact to contactless fingerprint matching. *IEEE Transactions on Information Forensics and Security*, 17:196–210, 2021.

[94] S. A. Grosz, J. J. Engelsma, N. G. Paulter, and A. K. Jain. White-box evaluation of fingerprint matchers: Robustness to minutiae perturbations. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2020.

[95] S. A. Grosz, J. J. Engelsma, R. Ranjan, N. Ramakrishnan, M. Aggarwal, G. G. Medioni, and A. K. Jain. Minutiae-guided fingerprint embeddings via vision transformers. *arXiv preprint arXiv:2210.13994*, 2022.

[96] S. A. Grosz and A. K. Jain. Spoofgan: Synthetic fingerprint spoof images. *IEEE Transactions on Information Forensics and Security*, 2022.

[97] S. A. Grosz and A. K. Jain. Afr-net: Attention-driven fingerprint recognition network. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2023.

[98] S. A. Grosz and A. K. Jain. Latent fingerprint recognition: Fusion of local and global embeddings. *IEEE Transactions on Information Forensics and Security*, 2023.

[99] T. Group. Biometrics: A comprehensive introduction. https://dis-blog.thalesgroup.com/identity-biometric-solutions/2017/01/30/biometricsintro/, 2017. Accessed: May 3, 2024.

[100] T. Group. Five james bond gadgets which use real-world technologies. https://dis-blog.thalesgroup.com/identity-biometric-solutions/2021/11/16/five-james-bond-gadgets-which-use-real-world-technologies/, 2021. Accessed: May 3, 2024.

[101] T. Group. Automated Fingerprint Identification System (AFIS) overview - A short history. https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/afis-history, 2023. Accessed: January 16, 2024.

[102] S. Gu, J. Feng, J. Lu, and J. Zhou. Latent fingerprint registration via matching densely sampled points. *IEEE Transactions on Information Forensics and Security*, 16:1231–1244, 2020.

[103] S. Gu, J. Feng, J. Lu, and J. Zhou. Latent fingerprint indexing: Robust representation and adaptive candidate list. *IEEE Transactions on Information Forensics and Security*, 17:908–923, 2022.

[104] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.

[105] M. R. Hawthorne. *Fingerprints: Analysis and Understanding*. CRC Press, 2009.

[106] 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*, pages 770–778, 2016.

[107] Z. He, E. Liu, and Z. Xiang. Partial fingerprint verification via spatial transformer networks. In *IEEE International Joint Conference on Biometrics*, pages 1–10. IEEE, 2020.

[108] E. R. Henry. *Classification and uses of finger prints*. HM Stationery office, 1922.

[109] W. J. Herschel. *The origin of finger-printing*. H. Milford, Oxford University Press, 1916.

[110] B. Y. Hiew, A. B. Teoh, and Y.-H. Pang. Touch-less fingerprint recognition system. In *2007 IEEE Workshop on Automatic Identification Advanced Technologies*, pages 24–29. IEEE, 2007.

[111] L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.

[112] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*, 2017.

[113] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[114] G. Huang and A. H. Jafari. Enhanced balancing gan: Minority-class image generation. *Neural Computing and Applications*, pages 1–10, 2021.

[115] J. Huang, W. Luo, W. Yang, A. Zheng, F. Lian, and W. Kang. Fvt: Finger vein transformer for authentication. *IEEE Transactions on Instrumentation and Measurement*, 2022.

[116] X. Huang, P. Qian, and M. Liu. Latent fingerprint image enhancement based on progressive generative adversarial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 800–801, 2020.

[117] International Standards Organization. ISO/IEC 30107-1:2016, Information Technology—Biometric Presentation Attack Detection—Part 1: Framework. https://www.iso.org/standard/53227.html, 2016.

[118] J. J Engelsma, D. Deb, A. Jain, A. Bhatnagar, and P. Sewak Sudhish. Infant-prints: Fingerprints for reducing infant mortality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 67–74, 2019.

[119] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[120] A. K. Jain, Y. Chen, and M. Demirkus. Pores and ridges: High-resolution fingerprint matching using level 3 features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):15–27, 2007.

[121] A. K. Jain and J. Feng. Latent fingerprint matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):88–100, 2010.

[122] A. K. Jain, S. Prabhakar, and S. Pankanti. On the similarity of identical twin fingerprints. *Pattern Recognition*, 35(11):2653–2663, 2002.

[123] J. Jang, S. J. Elliott, and H. Kim. On improving interoperability of fingerprint recognition using resolution compensation based on sensor evaluation. In S.-W. Lee and S. Z. Li, editors, *Advances in Biometrics*, pages 455–463, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[124] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-time Style Transfer and Super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[125] P. Johnson, F. Hua, and S. Schuckers. Texture modeling for synthetic fingerprint generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 154–159, 2013.

[126] I. Joshi, A. Anand, M. Vatsa, R. Singh, S. D. Roy, and P. Kalra. Latent fingerprint enhancement using generative adversarial networks. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 895–903. IEEE, 2019.

[127] K. M. Kenyon. *Archaeology in the holy land*. E. Benn; New York: WW Norton, 1979.

[128] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 54(10s):1–41, 2022.

[129] H. Kim, X. Cui, M.-G. Kim, and T. H. B. Nguyen. Fingerprint generation and presentation attack detection using deep neural networks. In *IEEE Conference on Multimedia Information Processing and Retrieval*, pages 375–378. IEEE, 2019.

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

[131] S. Kirchgasser, C. Kauba, and A. Uhl. Assessment of synthetically generated mated samples from single fingerprint samples instances. In *2021 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2021.

[132] S. Kirchgasser, C. Kauba, and A. Uhl. The plus multi-sensor and longitudinal fingerprint dataset: An initial quality and performance evaluation. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(1):43–56, 2021.

[133] P. D. Komarinski. Automated fingerprint identification systems. In *Cold Case Homicides*, pages 317–326. CRC Press, 2017.

[134] A. Kumar and Y. Zhou. Contactless fingerprint identification using level zero features. In *CVPR 2011 Workshops*, pages 114–119. IEEE, 2011.

[135] P. D. Lapsley, J. A. Lee, D. F. Pare Jr, and N. Hoffman. Anti-fraud biometric scanner that accurately detects blood flow, 1998. US Patent 5,737,439.

[136] C. Lee, S. Lee, and J. Kim. A study of touchless fingerprint recognition system. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 358–365. Springer, 2006.

[137] J. Li, J. Feng, and C.-C. J. Kuo. Deep convolutional neural network for latent fingerprint enhancement. *Signal Processing: Image Communication*, 60:52–63, 2018.

[138] K. Li and X. Yang. Diffusion probabilistic model based end-to-end latent fingerprint synthesis. In *2023 IEEE 4th International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 343–349. IEEE, 2023.

[139] R. Li, D. Song, Y. Liu, and J. Feng. Learning global fingerprint features by training a fully convolutional network with local patches. In *2019 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2019.

[140] R. Li, D. Song, Y. Liu, and J. Feng. Learning Global Fingerprint Features by Training a Fully Convolutional Network with Local Patches. *International Conference on Biometrics*, 2019.

[141] Z. Li, M. Cao, X. Wang, Z. Qi, M.-M. Cheng, and Y. Shan. Photomaker: Customizing realistic human photos via stacked id embedding. *arXiv preprint arXiv:2312.04461*, 2023.

[142] C. Lin and A. Kumar. Improving cross sensor interoperability for fingerprint identification. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 943–948. IEEE, 2016.

[143] C. Lin and A. Kumar. A cnn-based framework for comparison of contactless to contact-based fingerprints. *IEEE Transactions on Information Forensics and Security*, 14(3):662–676, 2018.

[144] C. Lin and A. Kumar. Contactless and partial 3d fingerprint recognition using multi-view deep representation. *Pattern Recognition*, 83:314–327, 2018.

[145] C. Lin and A. Kumar. Matching contactless and contact-based conventional fingerprint images for biometrics identification. *IEEE Transactions on Image Processing*, 27(4):2008–2021, 2018.

[146] C. Lin and A. Kumar. A cnn-based framework for comparison of contactless to contact-based fingerprints. *IEEE Transactions on Information Forensics and Security*, 14(3):662–676, 2019.

[147] M. Liu and P. Qian. Automatic segmentation and enhancement of latent fingerprints using deep nested unets. *IEEE Transactions on Information Forensics and Security*, 16:1709–1719, 2020.

[148] X. Liu, K. Raja, R. Wang, H. Qiu, H. Wu, D. Sun, Q. Zheng, N. Liu, X. Wang, G. Huang, et al. A latent fingerprint in the wild database. *IEEE Transactions on Information Forensics and Security*, 2024.

[149] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[150] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[151] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[152] L. Lugini, E. Marasco, B. Cukic, and I. Gashi. Interoperability in fingerprint recognition: A large-scale empirical study. In *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, pages 1–6. IEEE, 2013.

[153] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[154] D. Maio and D. Maltoni. Ridge-line density estimation in digital images. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, volume 1, pages 534–538. IEEE, 1998.

[155] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. Fvc2002. http://bias.csr.unibo.it/fvc2002/, 2002. 2002.

[156] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. Fvc2004. http://bias.csr.unibo.it/fvc2004/, 2004. 2004.

[157] A. Malhotra, A. Sankaran, A. Mittal, M. Vatsa, and R. Singh. Fingerphoto authentication using smartphone camera captured under varying environmental conditions. In *Human*

Recognition in Unconstrained Environments, pages 119–144. Elsevier, 2017.

[158] A. Malhotra, A. Sankaran, M. Vatsa, and R. Singh. On matching finger-selfies using deep scattering networks. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(4):350–362, 2020.

[159] A. Malhotra, M. Vatsa, R. Singh, K. B. Morris, and A. Noore. Multi-surface multi-technique (must) latent fingerprint database. *IEEE Transactions on Information Forensics and Security*, 2023.

[160] D. Maltoni, D. Maio, A. K. Jain, and F. Jianjiang. *Handbook of Fingerprint Recognition*. Springer Science & Business Media, 3 edition, 2022.

[161] E. Marasco, L. Lugini, B. Cukic, and T. Bourlai. Minimizing the impact of low interoperability between optical fingerprints sensors. In *IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2013.

[162] E. Marasco and A. Ross. A survey on antispoofing schemes for fingerprint recognition systems. *ACM Computing Surveys*, 47(2):28, 2015.

[163] E. Marasco and C. Sansone. On the Robustness of Fingerprint Liveness Detection Algorithms against New Materials used for Spoofing. In *Proc. Intl. Conf. Bio-Insp. Syst. Sign. Process.*, pages 553–558, 2011.

[164] E. Marasco and C. Sansone. Combining perspiration-and morphology-based static features for fingerprint liveness detection. *Pattern Recognition Letters*, 33(9):1148–1156, 2012.

[165] S. Marcel, M. S. Nixon, J. Fierrez, and N. Evans, editors. *"Handbook of Biometric Anti-Spoofing: Presentation Attack Detection"*. Springer, 2 edition, 2019.

[166] G. L. Marcialis, A. Lewicke, B. Tan, P. Coli, D. Grimberg, A. Congiu, A. Tidu, F. Roli, and S. Schuckers. First International Fingerprint Liveness Detection Competition—LivDet 2009. In *Proc. ICIAP*, pages 12–23. Springer, 2009.

[167] G. L. Marcialis, F. Roli, and A. Tidu. Analysis of fingerprint pores for vitality detection. In *Proc. 20th ICPR*, pages 1289–1292, 2010.

[168] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018.

[169] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino. Impact of artificial gummy fingers on fingerprint systems. In *Proc. SPIE*, volume 4677, pages 275–289, 2012.

[170] B. M. Mehtre and B. Chatterjee. Segmentation of fingerprint images—a composite method. *Pattern recognition*, 22(4):381–385, 1989.

[171] S. Minaee and A. Abdolrashidi. Finger-gan: Generating realistic fingerprint images using connectivity imposed gan. *arXiv preprint arXiv:1812.10482*, 2018.

[172] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang. Biometrics recognition using deep learning: A survey. *arXiv preprint arXiv:1912.00271*, 2019.

[173] V. Mistry, J. J. Engelsma, and A. K. Jain. Fingerprint synthesis: Search with 100 million prints. In *International Joint Conference on Biometrics*, 2020.

[174] A. A. Moenssens. *Fingerprint techniques*. Chilton Book Company London, 1971.

[175] V. Mura, L. Ghiani, G. L. Marcialis, F. Roli, D. A. Yambay, and S. A. Schuckers. LivDet 2015 - Fingerprint liveness detection competition 2015. In *Proc. IEEE 7th Intl. Conf. BTAS*, pages 1–6, 2015.

[176] V. Mura, G. Orrù, R. Casula, A. Sibiriu, G. Loi, P. Tuveri, L. Ghiani, and G. L. Marcialis. LivDet 2017 Fingerprint Liveness Detection Competition 2017. In *IEEE International Conference on Biometrics (ICB)*, pages 297–302. IEEE, 2018.

[177] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado. Fingerprint liveness detection using convolutional neural networks. *IEEE Transactions on Information Forensics and Security*, 11(6):1206–1213, 2016.

[178] ODNI, IARPA. IARPA-BAA-16-04 (Thor). https://www.iarpa.gov/index.php/research-programs/odin/odin-baa, 2016.

[179] U. D. of Homeland Security. Office of biometric identity management identification services. https://www.dhs.gov/obim-biometric-identification-services. Accessed on: January 16, 2024.

[180] F. B. of Investigation. Next generation identification (ngi) system fact sheet. https://le.fbi.gov/file-repository/december-2023-ngi-system-fact-sheet.pdf/view, 2023. Accessed on: January 16, 2024.

[181] U. S. F. B. of Investigation. *The Science of Fingerprints: Classification and Uses*. US Department of Justice, Federal Bureau of Investigation, 1984.

[182] A. Oig. Review of the fbi's handling of the brandon mayfield case. *Office of the Inspector General, Oversight and Review Division, US Department of Justice*, pages 1–330, 2006.

[183] K. Okereafor, I. Ekong, I. O. Markson, and K. Enwere. Fingerprint biometric system hygiene and the risk of covid-19 transmission. *JMIR Biomedical Engineering*, 5(1):e19623, 2020.

[184] OpenAI. Chatgpt. https://openai.com/blog/chatgpt, October 2022.

[185] OpenAI. Creating video from text. *OpenAI Blog*, February 2024.

[186] G. Orrù, R. Casula, P. Tuveri, C. Bazzoni, G. Dessalvi, M. Micheletto, L. Ghiani, and G. L. Marcialis. Livdet in action-fingerprint liveness detection competition 2019. In *International Conference on Biometrics*, pages 1–6. IEEE, 2019.

[187] H. İ. Öztürk, B. Selbes, and Y. Artan. Minnet: Minutia patch embedding network for automated latent fingerprint recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1627–1635, 2022.

[188] F. Pala and B. Bhanu. Deep Triplet Embedding Representations for Liveness Detection. In *Deep Learning for Biometrics. Advances in Computer Vision and Pattern Recognition.*, pages 287–307. Springer, 2017.

[189] C. T. Pang, Y. W. Yun, and J. Xudong. On-card matching. *Encyclopedia of Biometrics*, pages 1014–1021, 2009.

[190] S. Pankanti, S. Prabhakar, and A. K. Jain. On the individuality of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1010–1025, 2002.

[191] G. Parziale. Touchless fingerprinting technology. In *Advances in Biometrics*, pages 25–48. Springer, 2008.

[192] G. Parziale and Y. Chen. Advanced technologies for touchless fingerprint recognition. In *Handbook of Remote Biometrics*, pages 83–109. Springer, 2009.

[193] J. A. Pereira, A. F. Sequeira, D. Pernes, and J. S. Cardoso. A robust fingerprint presentation attack detection method against unseen attacks through adversarial learning. In *BIOSIG 2020-Proceedings of the 19th International Conference of the Biometrics Special Interest Group*. Gesellschaft für Informatik eV, 2020.

[194] J. Priesnitz, R. Huesmann, C. Rathgeb, N. Buchmann, and C. Busch. Mobile touchless fingerprint recognition: Implementation, performance and usability aspects. *arXiv preprint arXiv:2103.03038*, 2021.

[195] N. K. Ratha, S. Chen, and A. K. Jain. Adaptive flow orientation-based feature extraction in fingerprint images. *Pattern Recognition*, 28(11):1657–1672, 1995.

[196] A. Rattani, W. J. Scheirer, and A. Ross. Open set fingerprint spoof detection across novel fabrication materials. *IEEE Transactions on Information Forensics and Security*, 10 (11):2447–2460, 2015.

[197] A. Rattani, W. J. Scheirer, and A. Ross. Open set fingerprint spoof detection across novel fabrication materials. *IEEE Transactions on Information Forensics and Security*, 10(11):2447–2460, 2015.

[198] M. S. Riazi, S. M. Chavoshian, and F. Koushanfar. Synfi: Automatic synthetic fingerprint generation. *arXiv preprint arXiv:2002.08900*, 2020.

[199] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[200] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.

[201] A. Ross, S. Dass, and A. Jain. A deformable model for fingerprint matching. *Pattern Recognition*, 38(1):95–103, 2005.

[202] A. Ross, S. C. Dass, and A. K. Jain. Fingerprint warping using ridge curve correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):19–30, 2005.

[203] A. Ross and A. Jain. Biometric sensor interoperability: A case study in fingerprints. In D. Maltoni and A. K. Jain, editors, *ECCV Workshop BioAW*, pages 134–145. Springer, Springer, 2004.

[204] A. Ross and R. Nadgir. A calibration model for fingerprint sensor interoperability. In *Biometric Technology for Human Identification III*, volume 6202, page 62020B. International Society for Optics and Photonics, 2006.

[205] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann. Stabilizing training of generative adversarial networks through regularization. *Advances in Neural Information Processing Systems*, 30, 2017.

[206] P. C. Roy and V. N. Boddeti. Mitigating information leakage in image representations: A maximum entropy approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2586–2594, 2019.

[207] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[208] R. Saferstein. Criminalistics: An introduction to forensic science. 2004.

[209] A. Sankaran, T. I. Dhamecha, M. Vatsa, and R. Singh. On matching latent to latent fingerprints. In *2011 international joint conference on biometrics (IJCB)*, pages 1–6. IEEE, 2011.

[210] A. Sankaran, A. Malhotra, A. Mittal, M. Vatsa, and R. Singh. On smartphone camera

based fingerphoto authentication. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–7. IEEE, 2015.

[211] A. Sankaran, M. Vatsa, and R. Singh. Hierarchical fusion for matching simultaneous latent fingerprint. In *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 377–382. IEEE, 2012.

[212] A. Sankaran, M. Vatsa, and R. Singh. Multisensor optical and latent fingerprint database. *IEEE access*, 3:653–665, 2015.

[213] A. W. Senior and R. M. Bolle. Improved fingerprint matching by distortion removal. *IEICE Transactions on Information and Systems*, 84(7):825–832, 2001.

[214] A. Shoshan, N. Bhonker, E. Ben Baruch, O. Nizan, I. Kviatkovsky, J. Engelsma, M. Aggarwal, and G. Medioni. Fpgan-control: A controllable fingerprint generator for training with synthetic data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6067–6076, 2024.

[215] X. Si, J. Feng, J. Zhou, and Y. Luo. Detection and rectification of distorted fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):555–568, 2015.

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

[217] D. Song and J. Feng. Fingerprint indexing based on pyramid deep convolutional feature. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 200–207. IEEE, 2017.

[218] D. Song and J. Feng. Fingerprint Indexing based on Pyramid Deep Convolutional Feature. In *IEEE International Joint Conference on Biometrics*, pages 200–207. IEEE, 2017.

[219] D. Song, Y. Tang, and J. Feng. Aggregating minutia-centred deep convolutional features for fingerprint indexing. *Pattern Recognition*, 88:397–408, 2019.

[220] Y. Song, C. Lee, and J. Kim. A new scheme for touchless fingerprint recognition system. In *Proceedings of 2004 International Symposium on Intelligent Signal Processing and Communication Systems, (ISPACS)*, pages 524–527. IEEE, 2004.

[221] L. Spinoulas, H. Mirzaalian, M. E. Hussein, and W. AbdAlmageed. Multi-modal fingerprint presentation attack detection: Evaluation on a new dataset. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(3):347–364, 2021.

[222] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[223] Statista. Number of aadhaar cards generated in india. https://www.statista.com/statistics/1349391/india-number-of-aadhaar-cards-generated/, 2023. Accessed: May 3, 2024.

[224] C. Stein, C. Nickel, and C. Busch. Fingerphoto recognition with smartphone cameras. In *International Conference of Biometrics Special Interest Group (BIOSIG)*, pages 1–12. IEEE, 2012.

[225] G. Stragapede, P. Delgado-Santos, R. Tolosana, R. Vera-Rodriguez, R. Guest, and A. Morales. Mobile keystroke biometrics using transformers. *arXiv preprint arXiv:2207.07596*, 2022.

[226] K. Sundararajan and D. L. Woodard. Deep learning for biometrics: A survey. *ACM Computing Surveys (CSUR)*, 51(3):1–34, 2018.

[227] J. Svoboda, F. Monti, and M. M. Bronstein. Generative convolutional networks for latent fingerprint reconstruction. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 429–436. IEEE, 2017.

[228] E. Tabassi, M. Olsen, O. Bausinger, C. Busch, A. Figlarz, G. Fiumara, O. Henniger, J. Merkle, T. Ruhland, C. Schiel, and M. Schwaiger. NIST Fingerprint Image Quality 2. Technical Report NIST.IR.8382, National Institute of Standards and Technology, Gaithersburg, MD, 2021.

[229] B. Tan, A. Lewicke, D. Yambay, and S. Schuckers. The effect of environmental conditions and novel spoofing methods on fingerprint anti-spoofing algorithms. In *Proc. IEEE Intl. WIFS*, pages 1–6, 2010.

[230] S. Tandon and A. Namboodiri. Transformer based fingerprint feature extraction. *arXiv preprint arXiv:2209.03846*, 2022.

[231] S. Tang, C. Han, M. Li, and T. Guo. An end-to-end algorithm based on spatial transformer for fingerprint matching. In *7th International Conference on Computer and Communication Systems*, pages 320–325. IEEE, 2022.

[232] W. Tang, D. Figueroa, D. Liu, K. Johnsson, and A. Sopasakis. Enhancing fingerprint image synthesis with gans, diffusion models, and style transfer techniques. *arXiv preprint arXiv:2403.13916*, 2024.

[233] Y. Tang, F. Gao, and J. Feng. Latent fingerprint minutia extraction using fully convolutional network. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 117–123. IEEE, 2017.

[234] Y. Tang, F. Gao, J. Feng, and Y. Liu. Fingernet: An unified deep network for fingerprint minutiae extraction. In *IEEE International Joint Conference on Biometrics*, pages 108–116. IEEE, 2017.

[235] R. Tolosana, M. Gomez-Barrero, C. Busch, and J. Ortega-Garcia. Biometric Presentation Attack Detection: Beyond the Visible Spectrum. *IEEE Transactions on Information Forensics and Security*, 2019.

[236] R. Tolosana, M. Gomez-Barrero, J. Kolberg, A. Morales, C. Busch, and J. Ortega-Garcia. Towards Fingerprint Presentation Attack Detection based on Convolutional Neural Networks and Short Wave Infrared Imaging. In *IEEE International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5. IEEE, 2018.

[237] M. Trauring. Automatic comparison of finger-ridge patterns. *Nature*, 197(4871):938–940, 1963.

[238] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[239] B. T. Ulery, R. A. Hicklin, G. I. Kiebuzinski, M. A. Roberts, and J. Buscaglia. Understanding the sufficiency of information for latent fingerprint value determinations. *Forensic Science International*, 230(1-3):99–106, 2013.

[240] D. M. Uliyan, S. Sadeghi, and H. A. Jalab. Anti-spoofing method for fingerprint recognition using patch based deep learning machine. *Engineering Science and Technology, an International Journal*, 23(2):264–273, 2020.

[241] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[242] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[243] K. Wada. labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme, 2016.

[244] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[245] Q. Wang, X. Bai, H. Wang, Z. Qin, and A. Chen. Instantid: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*, 2024.

[246] C. I. Watson. NIST special database 14: Mated fingerprint cards pairs 2 version 2. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, 2001.

[247] C. I. Watson, G. P. Fiumara, E. Tabassi, S. L. Cheng, P. A. Flanagan, and W. J. Salamon.

Fingerprint vendor technology evaluation. 2015.

[248] C. I. Watson and C. L. Wilson. NIST special database 4. *Fingerprint Database, National Institute of Standards and Technology*, 17(77):5, 1992.

[249] R. Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[250] P. Wild, F. Daubner, H. Penz, and G. F. Domínguez. Comparative test of smartphone finger photo vs. touch-based cross-sensor fingerprint recognition. In *2019 7th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2019.

[251] A. B. V. Wyzykowski and A. K. Jain. Synthetic latent fingerprint generator. *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[252] A. B. V. Wyzykowski, M. P. Segundo, and R. de Paula Lemes. Level three synthetic fingerprint generation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9250–9257. IEEE, 2021.

[253] A. B. V. Wyzykowski, M. P. Segundo, and R. d. P. Lemes. Level three synthetic fingerprint generation. *arXiv preprint arXiv:2002.03809*, 2020.

[254] Q. Xie, Z. Dai, Y. Du, E. Hovy, and G. Neubig. Controllable invariance through adversarial feature learning. In *Advances in Neural Information Processing Systems*, pages 585–596, 2017.

[255] S. Yadav, C. Chen, and A. Ross. Relativistic discriminator: A one-class classifier for generalized iris presentation attack detection. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2635–2644, 2020.

[256] B. Yamashita and M. French. Fingerprint sourcebook-chapter 7: Latent print development. *US Dept. of Justice, Office of Justice Programs, National Institute of Justice*, 2010.

[257] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli, and S. Schuckers. LivDet 2011-Fingerprint liveness detection competition 2011. In *Proc. 5th IAPR ICB*, pages 208–215. IEEE, 2012.

[258] X. Yang, J. Feng, and J. Zhou. Localized dictionaries based orientation field estimation for latent fingerprints. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):955–969, 2014.

[259] X. Yin, Y. Zhu, and J. Hu. Contactless fingerprint recognition based on global minutia topology and loose genetic algorithm. *IEEE Transactions on Information Forensics and Security*, 15:28–41, 2019.

[260] S. Yoon, J. Feng, and A. K. Jain. Latent fingerprint enhancement via robust orientation field estimation. In *2011 international joint conference on biometrics (IJCB)*, pages 1–8. IEEE, 2011.

[261] S. Yoon, J. Feng, and A. K. Jain. Altered fingerprints: Analysis and detection. *IEEE TPAMI*, 34(3):451–464, 2012.

[262] S. Yoon and A. K. Jain. Longitudinal study of fingerprint recognition. *Proceedings of the National Academy of Sciences*, 112(28):8555–8560, 2015.

[263] B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

[264] J. Zhang, R. Lai, and C.-C. J. Kuo. Latent fingerprint segmentation with adaptive total variation model. In *2012 5th IAPR International Conference on Biometrics (ICB)*, pages 189–195. IEEE, 2012.

[265] J. Zhang, R. Lai, and C.-C. J. Kuo. Adaptive directional total-variation model for latent fingerprint segmentation. *IEEE Transactions on Information Forensics and Security*, 8(8):1261–1273, 2013.

[266] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[267] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.

[268] Q. Zhao, A. K. Jain, N. G. Paulter, and M. Taylor. Fingerprint image synthesis based on statistical feature models. In *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems*, pages 23–30. IEEE, 2012.

[269] Y. Zhong and W. Deng. Face transformer for recognition. *arXiv preprint arXiv:2103.14803*, 2021.

[270] W. Zhou, J. Hu, I. Petersen, S. Wang, and M. Bennamoun. A benchmark 3d fingerprint database. In *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 935–940, 2014.

[271] Y. Zhu, X. Yin, and J. Hu. Fingergan: A constrained fingerprint generation scheme for latent fingerprint enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.