CONTRIBUTIONS TO FINGERPRINT RECOGNITION

By

Joshua James Engelsma

A DISSERTATION

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

Computer Science – Doctor of Philosophy

2021

ABSTRACT

CONTRIBUTIONS TO FINGERPRINT RECOGNITION

By

Joshua James Engelsma

From the early days of the mid to late nineteenth century when scientific research first began to focus on fingerprints, to the present day fingerprint recognition systems we find deployed on our day to day devices, the science of fingerprint recognition has come a long way. In spite of this progress, there remains challenging problems to be solved. This thesis highlights a few of these problems, and proposes solutions to address them.

One area of further research that must be conducted on fingerprint recognition systems is that of robust, operational evaluations. In chapter two of this thesis, we show how the current practices of using calibration patterns to evaluate fingerprint readers are limited. We then propose a realistic fake finger called the Universal Target. The Universal Target is a realistic, 3D, fake finger (or phantom) which can be imaged by all major types of fingerprint sensing technologies. We show the entire manufacturing (molding and casting) process for fabricating the Universal Targets. Then, we show a series of evaluations which demonstrate how the Universal Targets can be used to operationally evaluate current commercial fingerprint readers. Our Universal Target is a significant step forward in enabling more realistic, standardized evaluations of fingerprint readers.

In our third chapter, we shift gears from improving the evaluation standards of fingerprint readers to instead focus on the security of fingerprint readers. In particular, we turn our attention towards detecting fake fingerprint (spoof) attacks. To do so, we open source a fingerprint reader (built from low-cost ubiquitous components), called RaspiReader. RaspiReader is a high-resolution fingerprint reader customized with both direct-view imaging and FTIR imaging in order to better detect fingerprint spoofs. We show through a number of experiments that RaspiReader enables state-of-the-art fingerprint spoof detection accuracy. We also demonstrate that RaspiReader enables better generalization to what are known as "unseen attacks" (those attacks which were not seen during training of the spoof detector). Finally, we show that fingerprints captured by

RaspiReader are completely compatible with images captured by legacy fingerprint readers for matching.

In chapter four, we move on to propose a major improvement to the fingerprint feature extraction and matching sub-modules of fingerprint recognition systems. In particular, we propose a deep network, called DeepPrint, to extract a 200 byte fixed-length fingerprint representation. While prevailing fingerprint matchers primarily utilize minutiae points and expensive graph matching algorithms for comparison, two DeepPrint representations can be compared with only 192 multiplications and 191 additions. This is extremely useful for large scale search where potentially billions of pairwise fingerprint comparisons must be made. The DeepPrint representation also enables practical encrypted matching using a fully homomorphic encryption scheme. This enables better protection of the fingerprint templates which are stored in the database. While discriminative fixed-length representations are available for both face and iris recognition, such a representation has eluded fingerprint recognition. This chapter aims to fill that void.

Finally, we conclude our thesis by working to extend fingerprint recognition to all ages. While current fingerprint recognition systems are being used by billions of teenagers and adults around the world, the youngest people among us remain disenfranchised. In particular, modern day fingerprint recognition systems do not work well on infants and young children. In this penultimate chapter, we aim to rectify this major shortcoming. To that end, we prototype a high-resolution (1900 ppi) infant fingerprint reader. Then, we track and fingerprint 315 infants (under the age of 3 months at enrollment) at the Dayalbagh Children's Hospital in Agra India over the course of 1 year (4 different sessions). To match the infant fingerprints, we develop our own high-resolution infant fingerprint matcher. Our experimental results demonstrate significant promise for the extension of fingerprint recognition to all ages. This work has the potential for major global good as all young infants and children could be given a verifiable digital identity for better vaccination tracking as a child and for government benefits and assistance as an adult.

In summary, this thesis makes major contributions to the entire end-to-end fingerprint recognition system and extends its use case to all ages.

To my family, my friends, and my loving wife Kate

ACKNOWLEDGMENTS

"Thy word is a lamp unto my feet, and a light unto my path."

Psalm 119:105

As I sit down to write this thesis and anticipate the end of my time as a PhD student, I sense a feeling of nostalgia already sweeping over me. Perhaps it is too cliche to say, but it nevertheless true, that the past five years have flown by. First and foremost, I am grateful to God for giving me the health, mind, and strength to complete this thesis. I am also incredibly grateful for all of the friends, colleagues, and mentors who have instructed me, guided me, and learned along side of me during this journey. Although there are too many to adequately address in this confined space, I want to call out a few individuals who have been particularly special to me throughout the PhD.

I want to begin by thanking my advisor Dr. Jain. When I arrived in East Lansing five years ago, I was an eager but raw PhD student that lacked any significant research training. In spite of this, Dr. Jain took a chance on me in bringing me into his lab and patiently teaching me how to rigorously conduct and disseminate scientific research. When I look back to what I was as a researcher when I started to where I am today, I can see remarkable growth. I attribute this to the guidance and example of Dr. Jain. Dr. Jain has told me several times that *"being a PhD advisor is like polishing a cloudy diamond"*. Thank you Dr. Jain for your hard work in polishing this cloudy diamond. Thank you also for your friendship and timely counsel on matters outside of research throughout the PhD.

Next, I want to thank my wife and best friend Kate. Kate has selflessly moved away from family and friends to facilitate my education and continues to tirelessly support me and our two children, Vivian and Abigail, in so many ways. During the most difficult times of the PhD, her support and encouragement kept me moving forward. She deserves the credit for this PhD as much as I do. Kate, you, Vivian and Abigail bring me unspeakable joy in life.

Thank you to all of my family. I am especially thankful to my parents for raising me, providing for me, and encouraging me to chase big dreams. Thank you all for the numerous one hour trips from Grand Rapids to Lansing for family fellowship. Thank you Charlie and James for meeting me out here on occasion for much needed study reprieve via bike rides and video games.

I am grateful for all of my fellow PRIPies, especially the members of my cohort (Debayan, Yichun, and Sixue). I want to especially thank Deb for his close friendship throughout the PhD. I have many fond memories of playing pickup soccer together, attending Hindi musical events, debating about current events over a beer, and traveling to India for research on five separate occasions. Thank you Sixue and Yichun for all of our conversations in the lab, your encouragements throughout, and for your willingness to collaborate with me on coursework and research. Thank you also to Steven Grosz, Vishesh, Divyansh, Inci, Tarang, Cori, and Sunpreet, current and former PRIPies with whom I had the pleasure of studying with. Thank you also to non-PRIPies Yaojie, Amin, Joel, Renu, Cunjian, Steven Hoffman, and Adam Terwilliger for your friendship. A special thanks to Kai Cao, a PRIP post-doc whose mentorship has been invaluable to me and who has become a close friend. Finally, thanks to all of those involved on the various IARPA ODIN GCTs. I have many fun memories of our travels to the Baltimore area together.

Thank you to my committee, Dr. Xiaoming Liu, Dr. Arun Ross, and Dr. Mi Zhang for your feedback and comments on this thesis. A special thanks to Dr. Liu and Dr. Ross for your feedback throughout the IARPA ODIN project and your general life advice. Thank you Dr. Vishnu Boddeti for your collaboration with me on encrypted matching. Thank you Brenda Hodge, Steven Smith, and Amy King for your administrative assistance and your patience with my tardy travel reimbursement forms. A special thanks to Christopher Perry, for support on the IARPA ODIN project and for being a great friend, Brian Wright for his assistance with the ECE department 3D printer, and Dr. Anjoo Bhatnagar and Dr. Prem Sudhish for their facilitating our infant data collections in India. Finally, I would like the thank NIST (especially Nick Paulter), and IARPA for their support of the research contained in this thesis. There are of course many others who have impacted me along the way and I thank all of you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction

When many of us hear the words *"fingerprint"* or *"fingerprint recognition"*, our minds will immediately wander to a number of science fiction or crime solving television shows such as *Person of Interest*[1], *NCIS*[2], or *Forensic Files*[3], where a fingerprint left behind at a crime scene is used to identity a suspected criminal, or, in the case of *Person of Interest*, is used to spoof a fingerprint recognition system with a 3D printed fingerprint in order to gain access to a secure facility. The prevalence of fingerprints and fingerprint recognition in modern day entertainment shows is a testament to the ever increasing ubiquity of fingerprint recognition in our society. Indeed, fingerprint recognition systems are now widely deployed across a plethora of different applications including government services and facility access, smartphone unlock, forensics, customs and border control, and national ID [111].

In this chapter, we explain how fingerprint recognition came to be so pervasive in our day to day lives. We begin by discussing the early history of fingerprints and their progression (through major scientific advances) towards the many applications, both in law enforcement as well as consumer applications, we find them in today. We then describe the pipeline of a standard fingerprint recognition system along with all of its constituent modules. Finally, we list some of the challeng-

[1] https://www.imdb.com/title/tt1839578/

[2] https://www.imdb.com/title/tt0364845/

[3] https://www.imdb.com/title/tt0247882/

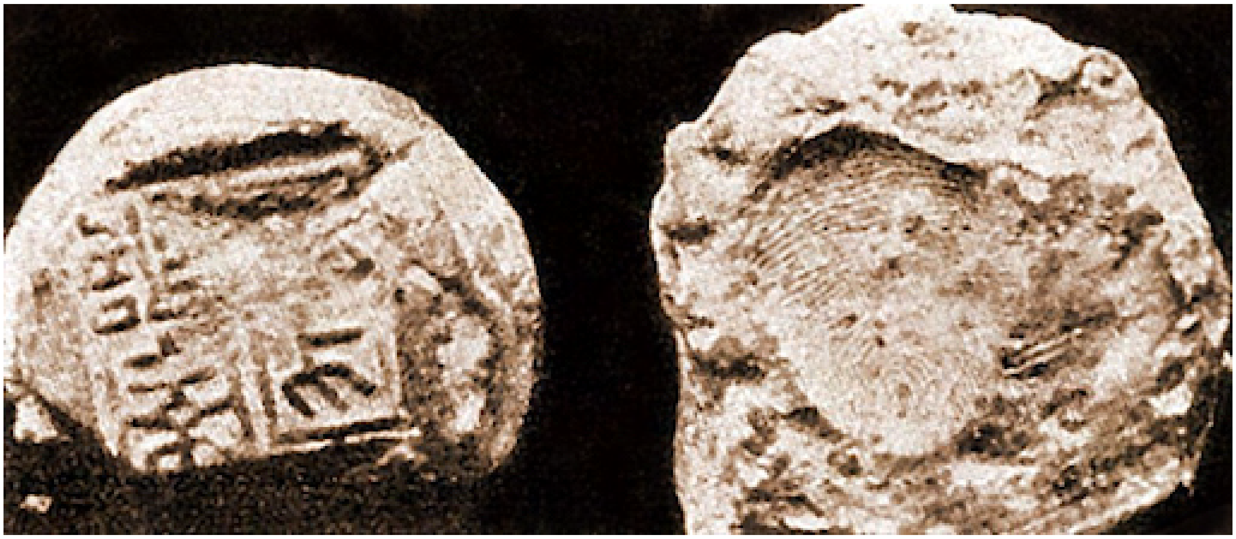


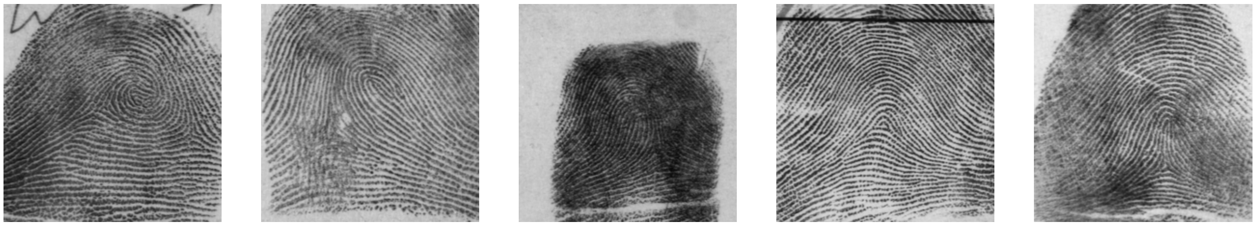
Figure 1.1 One of the earliest recorded uses of fingerprints include chinese clay seals which were used to sign business transactions. Image retrieved from [63].

ing problems in state-of-the-art fingerprint recognition systems and give an overview of how this thesis aims to address these limitations.

1.1 History of Fingerprint Recognition

Human interest in fingerprints dates back thousands of years. In fact, early examples of fingerprint patterns have been found in ancient Babylonian tablets dated back to 1955-1913 BC and later on in ancient Chinese clay seals dated 600-700 AD [111]. However, it was not until centuries later that fingerprint recognition came to be studied with systematic scientific rigor for person recognition.

In 1684, Nehemiah Grew published the first scientific study on the ridges, furrows, and pore structure of fingerprints [111]. Following this first scientific paper, the late 19^{th} century saw several prominent scientists make major contributions to the field of fingerprint recognition. Sir William Herschel first proposed that the ridge structure of the fingerprint remained unaltered over time by examining his fingerprint pattern in 1860 and then again in 1890 [35]. 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 [35]. In other words, fingerprints were a *permanent* physical characteristic of an individual that remained with them throughout their



(a) Whorl (b) Right Loop (c) Left Loop (d) Arch (e) Tented Arch

Figure 1.2 Example fingerprints of each of the five major fingerprint classes defined by Sir Edward Henry. Images retrieved from the NIST SD4 database [129].

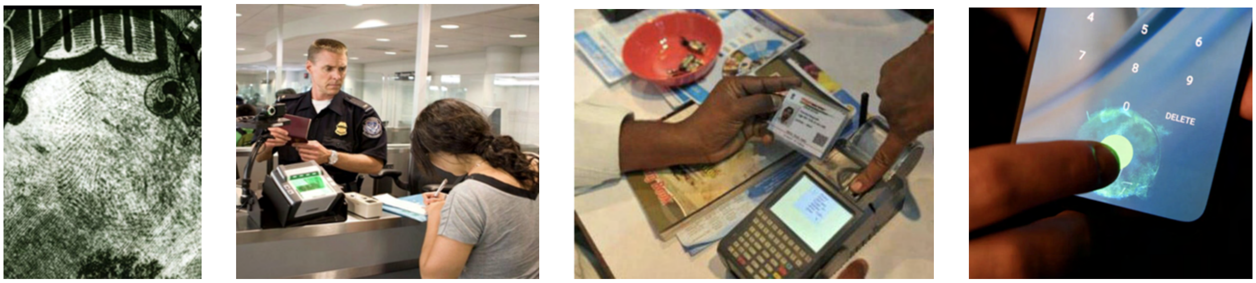
lifetime. This characteristic of fingerprint permanence remains a central tenant to modern day fingerprint recognition systems [92, 111].

Other notable pioneers include Sir Edward Henry, and Sir Francis Galton. Henry is credited with devising the Henry Classification system which categorizes fingerprints into one of several classes (Figure 1.2). This classification system was adopted by the New Scotland Yard in 1901 for criminal identification [35]. Sir Francis Galton, a polymath and cousin of Charles Darwin, wrote the landmark book *"Finger Prints"* in 1892. In it, he wrote down what he observed made fingerprints *unique* and thus able to identify an individual [35]. In particular, Galton made note of fingerprint minutiae, the endings and bifurcations throughout the fingerprint ridge structure, which are still the most common feature representation used in automated fingerprint identification systems today [60].

Following the seminal studies of fingerprints in the late nineteenth century, a paper detailing the first *algorithmic* approach to fingerprint recognition was proposed by Mitchell Trauring in Nature, 1963 [170]. In the 60 years since this first paper on automated fingerprint recognition was published, significant progress has been achieved in the area of automated fingerprint identification (AFIS) systems. Today, standardized evaluations such as NIST FpVTE 2012 [178] and the more recent FVC-ongoing [39, 107], show fingerprint authentication accuracies (single finger) as high as $TAR = 99.98\%$ @ $FAR = 0.01\%$ (FVC-ongoing standard 1-to-1 database) and $TAR = 99.4\%$ @ $FAR = 0.01\%$ (FVC-ongoing hard 1-to-1 database). Fusing the scores from multiple fingers can even further boost the recognition performance which is why fingerprints from all 10 fingers are captured by law enforcement agencies and national ID systems. The cause of remaining failure cases can be attributed to noisy and heavily distorted fingerprint images [178].

In addition to becoming nearly perfect with respect to recognition accuracy, modern day AFIS are much more well understood thanks to rigorous statistical studies on the central tenants of fingerprint recognition over the last 20 years. In particular, it had long been assumed to be true that fingerprints were both (i) **unique** (to a specific finger and a specific person), and (ii) **permanent** (they do not change over time) without a strong statistical backing. This changed for the better in 2002 when Pankanti, Prabhakar, and Jain published their study: *"On the individuality of fingerprints"* [136]. In it, the authors developed statistical models that showed that the probability of two fingerprint patterns, each with 36 minutiae would have a probability of $5.47 \times\ 10^{-59}$ of sharing all 36 minutiae points. In other words, the probability of two identical fingerprints was incredibly miniscule. Following this statistical study on the individuality or uniqueness of fingerprints, in 2015, Yoon and Jain published another study in the Proceedings of the National Academy of Sciences (PNAS) providing strong backing to the premise that fingerprints are permanent, or remain the same over time [188]. In their study, Yoon and Jain analyzed automated fingerprint match scores from 15,597 subjects over time lapses of 5-12 years. They found that although the match scores for a given subject did drop over time, the overall recognition accuracy remained stable for the maximum time interval in the dataset of 12 years. This lends strong statistical evidence to the premise that fingerprints are indeed a permanent physical trait.

From the early days of manual fingerprint comparison to the modern day highly accurate AFIS, the study of fingerprints have come a long way. Not surprisingly, this has led to a proliferation of fingerprint recognition systems into a number of different applications globally. In the next section, we discuss some of the more well known of these fingerprint recognition applications.



(a) Forensics (b) Border Security (c) National ID (d) Mobile Unlock

Figure 1.3 Various applications of fingerprint recognition. (a) An example of a latent fingerprint left behind on a dollar bill, which could be subsequently used to search a database of known criminals; (b) a woman has her fingerprints taken at US Customs (OBIM system) prior to entry into the country; (c) An Indian citizen is authenticated by the Aadhaar system; (d) a mobile phone is unlocked, bypassing the need for a password or key-code for access. Images retrieved from Google Images.

1.2 Major Applications

Thanks to the high performance (accuracy and speed) of modern day AFIS [39], and the strong statistical backing of the foundational premises (**uniqueness** and **permanence**) upon which AFIS are built, fingerprint recognition has exploded into a myriad of different applications throughout our world today. Some of the more noteworthy or well known of these applications are enumerated below and are shown in Figure 1.3.

- **Forensics:** Already in the mid to late nineteenth century, Faulds, Herschel, Henry, and Bertillon were manually examining fingerprints to identify repeat criminals [35]. In 1924, the FBI formally started the Identification Division of the FBI to collect and store inked ten-print cards from criminals. Later on in 1999, this was rolled into the FBI's Integrated Automated Fingerprint Identification System (IAFIS) where fingerprints (tenprints) were digitized, stored, and automatically compared. Finally, in 2011, the FBI's Next Generation Identification (NGI) system was established to improve upon the outdated IAFIS system [56]. In particular, it enabled faster and more accurate fingerprint recognition capabilities and utilized additional biometric modalities (face and iris). Today, NGI continues to maintain a database of 78 million criminal fingerprint records, and 58 million civilian fingerprint records. In August of 2020, the NGI system averaged 18,000 suspected criminal latent fingerprint queries per day [57].

- **Border Security:**

  The Office of Biometric Identity Management (OBIM, formerly US-VISIT) program manages the largest biometric repository in the United States. In 2020, the program was projected to process 152 million query records against a continually growing gallery size of 280 million records. The OBIM vision statement is to "lead the use of biometric identity for a safer world, enhanced individual privacy, and improved quality of life" [37]. One of the primary ways OBIM accomplishes this vision statement is by preventing criminals and dangerous individuals from entering the United States. Of the 88 million query transactions recorded in 2014, OBIM successfully flagged 2.7 million queries to individuals on the United States watchlist[4].

- **National ID:**

  The world's largest biometric recognition system is India's Aadhaar (meaning *foundation* in Hindi). Aadhaar uses all 10 fingerprints, 2 irises, and face image of an Indian citizen to deduplicate and then link the citizen to a 12-digit unique identifier. As of September of 2020, over 1.2 billion Indian citizens have been enrolled into Aadhaar[5]. This system is successfully used to provide benefits to the marginalized segment of the population and to facilitate financial transactions. One limitation of Aadhaar is that it starts enrollment at the age of 5 years. Unfortunately, this leaves many of India's more vulnerable citizens (its infants and children) at risk.

- **Mobile Unlock and Payments**

  Perhaps the most prevalent use of fingerprint recognition in today's world is that of mobile unlock and payments. As of August 2020, there are estimated to be 5.15 billion smartphone

[4] https://bit.ly/3cCSEgL

[5] https://uidai.gov.in/aadhaar_dashboard/

users globally[6]. Of these over 5 billion smartphones, it has been reported that in 2018, 60% of them would be equipped with fingerprint recognition technology for unlock and payment services[7]. Furthermore, this percentage is on a year over year upward trend.

Due to the convenience fingerprint recognition technology has afforded smartphones, major payments companies (Visa[8] and Master Card[9]) are now integrating fingerprint recognition technologies directly into credit cards via a concept referred to as "Match on Card". Match on Card would enable a user to enroll their fingerprint template to a chip on their credit card which could then be used to perform a financial transaction in lieu of a pin number. The fingerprint template data would never leave the credit card (so as to keep the fingerprint template secure) as it would be directly matched to the query fingerprint on the chip on the card.

The applications and statistics enumerated above indicate that it is entirely plausible that over half of the world's population are now using fingerprint recognition in their day to day lives. The data also suggests that these numbers will continue to grow. In the next section, we dive into the pipeline of modern day automated fingerprint recognition systems to better understand (from a technical point of view) how fingerprint recognition systems have become so prevalent in our day to day lives.

1.3 Pipeline of Fingerprint Recognition Systems

Automated fingerprint recognition systems operate in one of three major modes: (i) enrollment, (ii) authentication, or (iii) search. Each of these modes of operation is supported by multiple sub-modules within the recognition system including fingerprint (1) sensing, (2) feature extraction, and (3) matching. In the following section, we describe each of these modes of operation, and also the individual sub-modules that support such functionality.

[6] https://bit.ly/2HyRojs

[7] https://bit.ly/3cATAlH

[8] https://vi.sa/3mUT3A8

[9] https://bit.ly/3j7lJn9

- **Enrollment:** During the enrollment stage (Figure 1.4), a fingerprint is captured by the fingerprint reader and transferred to the feature extractor. The feature extractor then extracts salient, discriminative features (*e.g.* minutiae points) and packs them into a template along with the user's meta-data. Finally, the template is saved in the enrollment database. Ideally, this template should be encrypted prior to its storage in the template database to protect against the event that a hacker breaks into the enrollment database. After enrollment, the fingerprint recognition system can operate in one of two modes. It can be utilized for authentication (1:1 matching) or search (1:N matching) [111].

- **Authentication:** Fingerprint authentication refers to a 1 to 1 or (1:1) matching application. In such a scenario, the user presents their fingerprint to the reader along with a claimed identity (*e.g.* a PIN, or the identity is implicitly known via ownership of a device). Subsequently, a probe feature set (or template[10]) is extracted from the newly presented fingerprint, and an enrollment template for the claimed identity is retrieved from the database. Finally, these two templates are compared to make the binary decision of match or no match (Figure 1.5). Examples of fingerprint authentication include access control, and India's Aadhaar, where authentication is made based upon a 12-digit Aadhaar number for government benefits and assistance [111].

- **Search:** Fingerprint search refers to a 1 to N or $(1 : N)$ matching application (where N is the number of all the users in the enrollment database). In this scenario, a user's fingerprint is again captured by the fingerprint reader. This query fingerprint is then passed to the feature extractor to extract a salient, discriminative template. This query template is then matched to each of the N templates already enrolled in the gallery. The matcher returns a rank order list of possible candidates which are most similar to the query or probe representation (Figure 1.6). A notable example of fingerprint search is when a fingerprint left behind at a crime scene (or a latent fingerprint) is searched against a criminal database

[10] We refer to the terms *feature set*, *template*, *representation* interchangeably in this thesis.

in an effort to identity a suspect [111]. Another example is the de-duplication done prior to enrollment in a national ID system such as Aadhaar.

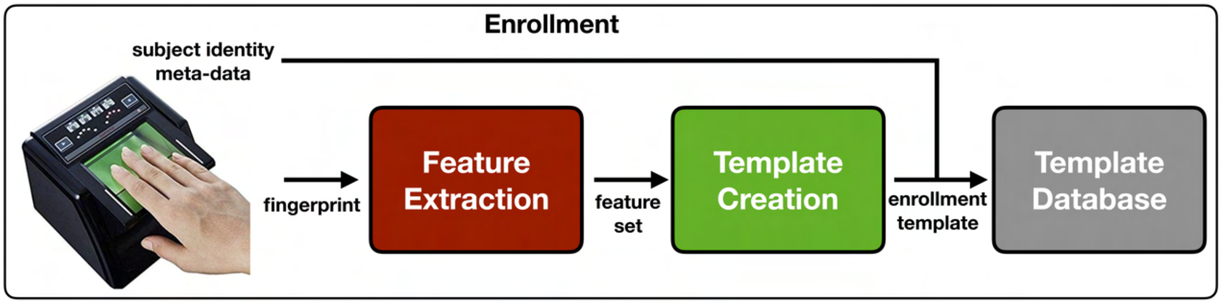


Figure 1.4 Enrollment Phase. A fingerprint is captured by the reader and transferred to the feature extractor where minutiae and other salient, discriminative features are extracted and packed into a template. The extracted template is then stored in the database.

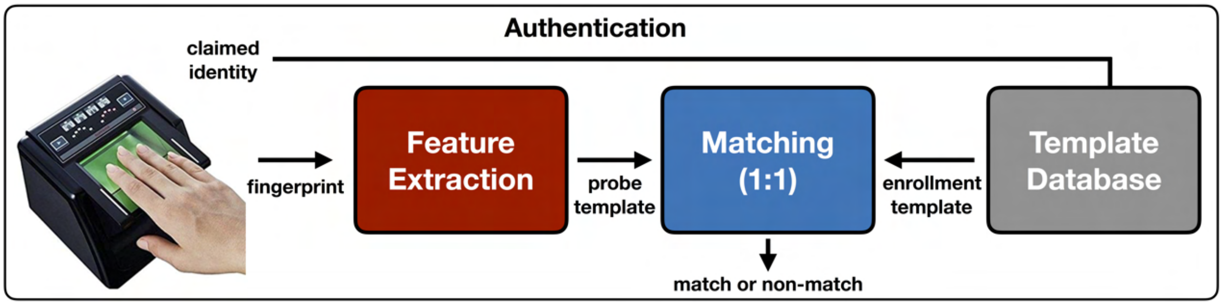


Figure 1.5 Fingerprint Authentication Schematic. During authentication, a 1:1 match is conducted between a newly extracted template, and a template already stored in the database. In this scenario, we are answering the question, *"Is this person a match to the specified enrollment template?"*

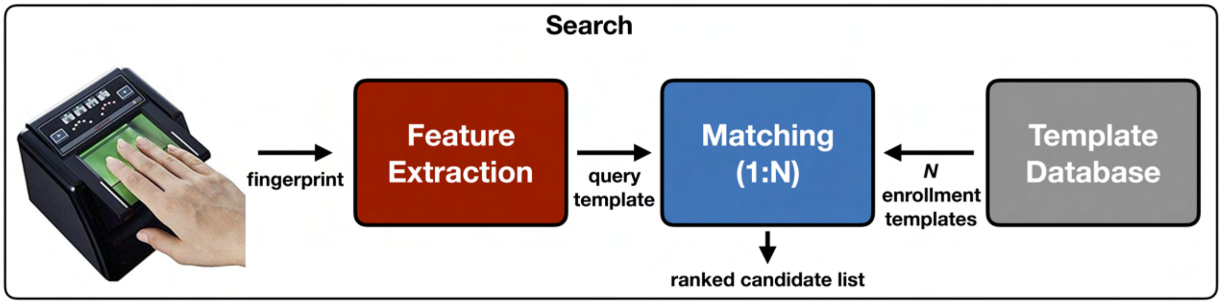


Figure 1.6 Fingerprint Search Schematic. During search, N matches are conducted between a newly extracted template (probe or query), and N templates already stored in the database. The matcher returns a ranked list of the candidates most similar to the query. In this scenario, we are answering the question, *"Who is this person?"*

The functionality of each of the aforementioned fingerprint recognition modes of operation is enabled by a configuration of multiple sub-modules (fingerprint reader, feature extractor, matcher) within the fingerprint recognition system. Each of these sub-modules are described below.

1.3.1 Fingerprint Readers

Early applications of fingerprint recognition in law enforcement required inking a user's fingers, and having them press down on a sheet of card stock paper (Fig 1.7). The fingerprints captured could be rolled fingerprints (captured 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. Since that time, a number of fingerprint readers have been developed which are significantly more convenient than the old "ink on paper" capture techniques.

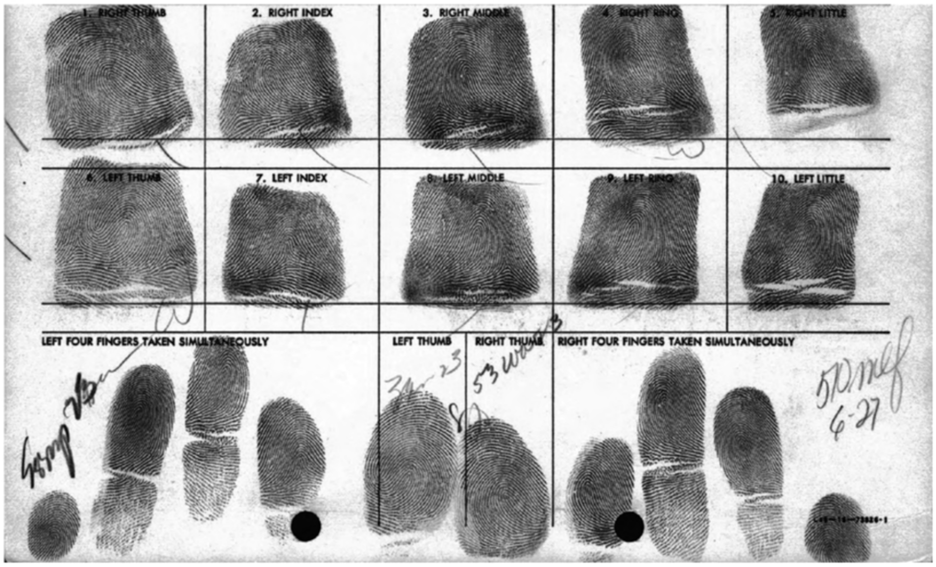


Figure 1.7 Fingerprints captured using ink and card stock paper. The fingerprints in the top row are rolled fingerprints, whereas the fingerprints in the bottom row are slap/plain fingerprints. Image reproduced from [88].

Fingerprint readers or scanners use a variety of sensing technologies to capture and convert a physical fingerprint into a digital image. We make a distinction in this thesis between the terms *reader* and *sensor* since a reader technically is utilizing one of a number of different sensors to capture the digital image. The major types of these sensing technologies include ultrasound, frustrated total internal reflection (FTIR), direct-view imaging, capacitance, thermal, and pressure sensors. Most all of these readers capture fingerprints at 500 pixels per inch (ppi). More expensive readers can capture at higher resolution (*e.g.* 1000 ppi) in order to capture finer level features [111]. More details on individual sensing technologies are enumerated below:

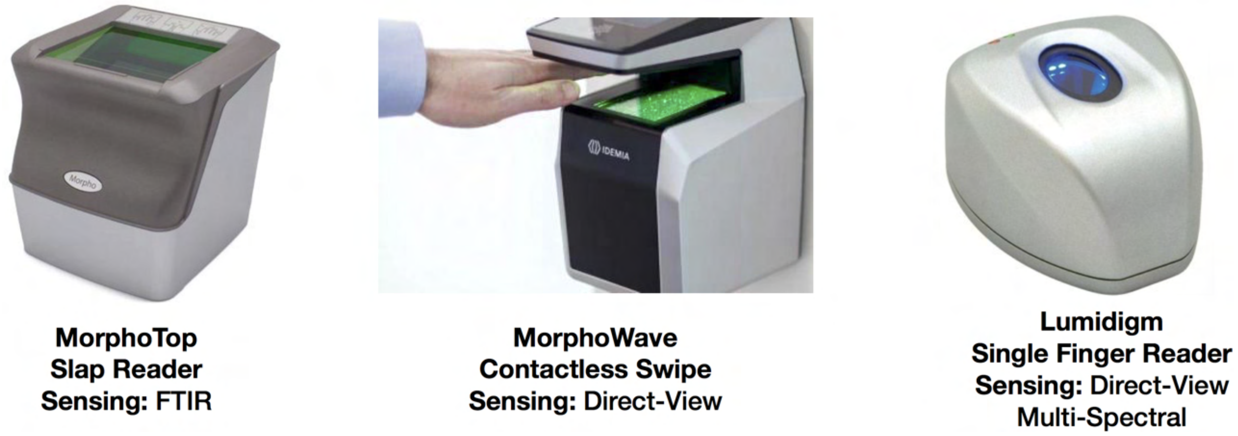
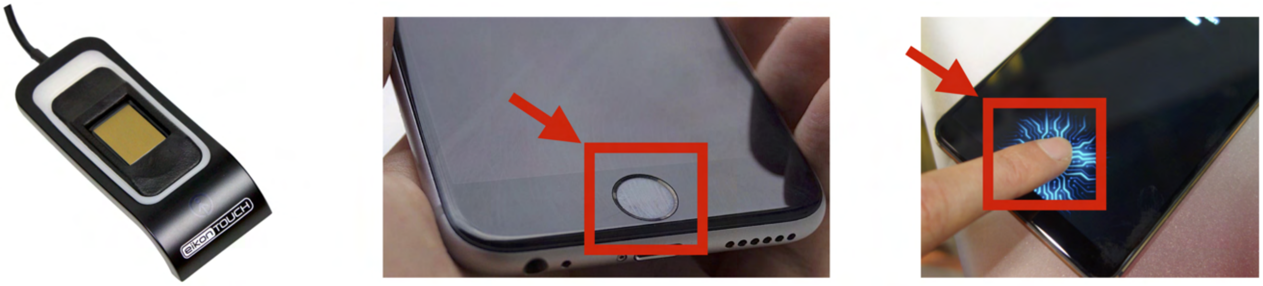


Figure 1.8 Examples of different types of optical-based fingerprint readers (Idemia *https://www.idemia.com/*, HIDGlobal *https://www.hidglobal.com/*). Images retrieved from Google Images.

- **Optical:** Of all the optical sensing technologies (Fig. 1.8), the most widely utilized is that of frustrated total internal reflection (FTIR). FTIR sensing works by making use of both a light source and a glass prism. In particular, by mounting a camera at an appropriate angle to a glass prism, light from the fingerprint ridges in contact with the glass prism are reflected back to the camera, while light from the valleys scatter [111]. The result is a high contrast fingerprint image.

  Other common optical fingerprint readers use direct-view imaging where a light source illuminates the finger and light from both the ridges and valleys are reflected back towards the camera [111]. The Lumidigm multi-spectral reader is built upon this sensing technology. Many contactless optical readers (*e.g.* the Morpho Wave) also use direct-view imaging. Direct-view fingerprints have lower contrast than FTIR fingerprints, but are less impacted by moisture on the finger due to environmental humidity.



Eikon Touch
Single Finger Reader
Sensing: Solid State

iPhone Touch ID
Single Finger Reader
Sensing: Solid State

Galaxy s10 in Display
Single Finger Reader
Sensing: Ultrasound

Figure 1.9 Examples of different types of solid-state readers (both capacitive) and an in-display, mobile-phone, ultrasound reader. Images retrieved from Google Images.

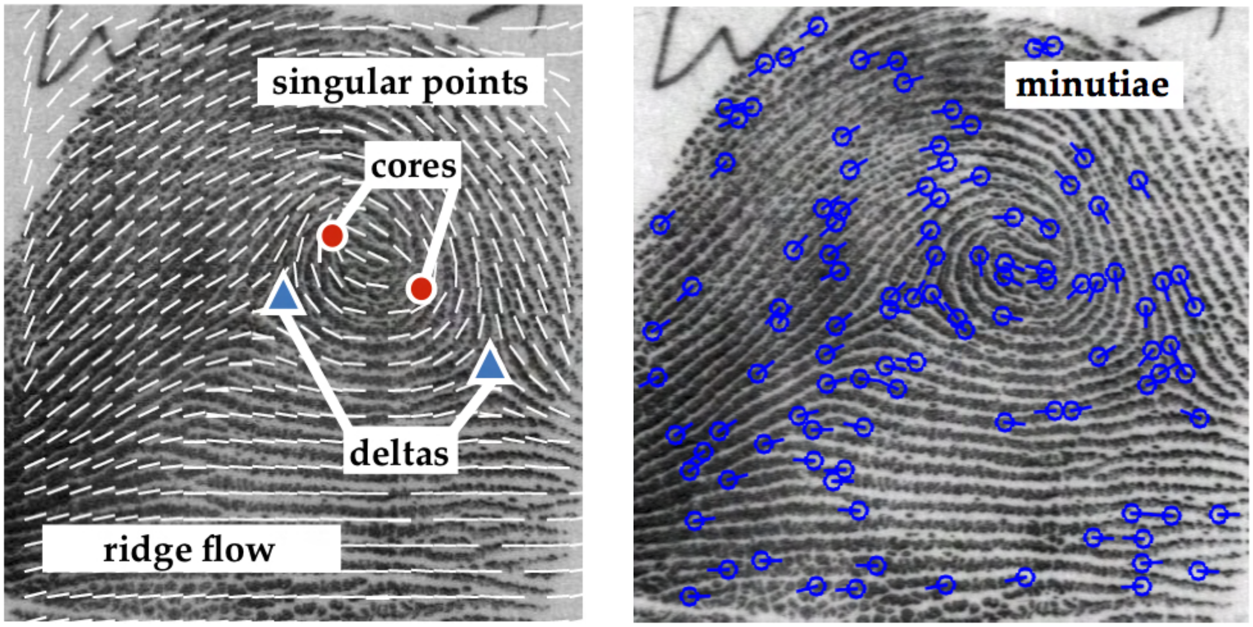
- **Solid-State:**

  Solid state sensing technology (Fig. 1.9) operates by using an array of mini-sensors which measure differentials in capacitance, temperature, or pressure between the ridges and valleys. Because of their small size and low cost, solid state sensors are the most commonly deployed sensing technology on mobile devices [111].

- **Ultrasound:**

  Ultrasound sensing (Fig. 1.9) works by employing acoustic waves towards the fingertip on the platen. Then, a receiver gathers the echoed responses and develops a depth profile of the fingerprint [111]. One of the main advantages of ultrasound sensing is that it enables a sub-surface fingerprint image to be captured. This sub-surface fingerprint could be a useful for detecting fake fingerprint attacks (otherwise commonly known as spoof attacks or presentation attacks). The sub-surface fingerprint is also particularly useful for improving the image quality of the elderly population which often has worn out and damaged fingerprints on the surface of the finger, but will still have a higher quality sub-surface fingerprint. Up until recently, commercial ultrasound readers held a very minor marketshare. However, recently QualComm Inc. developed an in display ultrasound sensor for the mobile phone. This sensor has been widely deployed in the Samsung smartphone series (Galaxy S10 onwards) [11].

[11] https://www.samsung.com/global/galaxy/what-is/ultrasonic-fingerprint/



(a) Level-1 features

(b) Level-2 features

Figure 1.10 The most popular fingerprint representation consists of (a) global level-1 features (ridge flow, core, and delta) and (b) local level-2 features, called minutiae points, together with their descriptors (e.g., texture in local minutiae neighborhoods). The fingerprint image illustrated here is a rolled impression from the NIST SD4 database [129]. The number of minutiae in NIST4 rolled fingerprint images range all the way from 12 to 196.

Two chapters of this thesis will focus heavily on the fingerprint reader component of fingerprint recognition systems. In particular, (i) the second chapter of this thesis involves the proper operational evaluation of the aforementioned fingerprint readers, particularly evaluating the interoperability of fingerprint recognition systems when two different types of sensing technologies are used to capture the enrollment image and the probe/query image. (ii) Chapter 3 of this thesis focuses on securing the fingerprint reader module by preventing fake fingerprint attacks (more commonly referred to as spoof attacks or presentation attacks).

1.3.2 Feature Extraction

After acquiring a digital format of the fingerprint via a fingerprint reader employing one of the aforementioned sensing technologies, the next step in the fingerprint recognition pipeline is to extract salient and discriminative features which will comprise the fingerprint template. Fingerprint feature sets are usually separated into one of three levels of features (Fig. 1.10).

- **Level-1:** Level-1 features are coarse or more global level features (Fig. 1.10). These features include the orientation field (or ridge-flow), and ridge spacing statistics of the fingerprint image. Additionally, major landmarks called singularities (further divided into core points and deltas) are included. Finally, the fingerprint type (earlier introduced as the Henry Classification System in Figure 1.2) can be considered as Level-1 features. While Level-1 features can be useful for aligning fingerprints, quickly indexing large galleries for a candidate list, or classifying them in accordance with the Henry Classification System, they are not discriminative enough for recognition standalone [111].

- **Level-2:**

  Level-2 features are more local features than the global Level-1 features (Fig. 1.10). These features are known as minutiae points. A minutiae point can be one of two types. Either it is a ridge bifurcation (a point at which a running ridge splits into two), or an ending (a point at which a running ridge terminates) (Fig. 1.11). Furthermore, each minutiae is comprised of a spatial location in the fingerprint $\{x, y\}$ and a ridge orientation (θ). The collection of all minutiae in a fingerprint image comprise the most widely standardized and utilized fingerprint template. Minutiae points are best extracted at a fingerprint image resolution 500 pixels per inch (ppi) [111]. They can be extracted using a variety of different techniques including the more recent deep network based approaches [125, 168].

- **Level-3:**

  Level-3 features are the finest level of fingerprint features (Fig. 1.11). They include features such as sweat pores, dots, and incipient ridges. In order to capture Level-3 features, a more expensive 1000 ppi fingerprint reader is required [86]. Due to the high-resolution requirement, and computational cost of extraction, Level-3 features are typically not used for fingerprint recognition even though it has been shown that they can be used to further

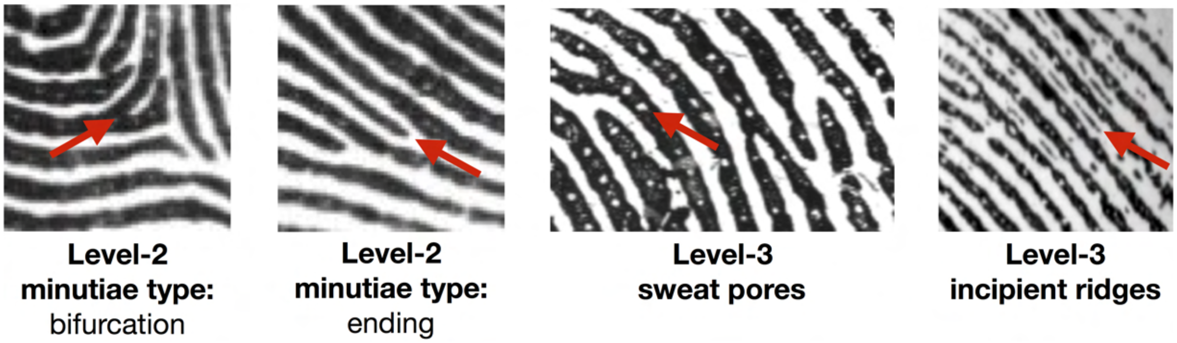


Figure 1.11 Examples of Level-2 features (two types of minutiae) and Level-3 features (sweat pores and incipient ridges).

improve the recognition performance [111]. Several potential use cases for Level-3 features which can afford the computational complexity tradeoff in favor of higher accuracy include high-resolution infant fingerprint matching and latent fingerprint matching.

More recently, deep networks have been employed to extract deep features from fingerprint images, including fixed-length fingerprint representations comprised of textural and minutiae related features [48]. Indeed, Chapter 4 of this thesis demonstrates the use of a deep network, called DeepPrint to extract a fixed-length representation from a fingerprint image to enable faster large scale search and more secure matching within the encrypted domain.

1.3.3 Template Database

A fingerprint template is comprised of some combination of the aforementioned feature sets, along with a user's meta-data. The International Standards Organization (ISO) has defined a standard template under ISO/IEC 19794-2 (2005)[12] (essentially a minutiae set). Typically, a commercial vendor will support extraction of the ISO standard template (to enable compatibility and interoperability with legacy template databases) and an additional proprietary template comprised of other levels and types of features for improved accuracy and speed.

The collection of templates from all subjects comprise the template database. Due to the plethora of personal identifying information (PII) in the template database, the template database

[12] https://www.iso.org/standard/52537.html

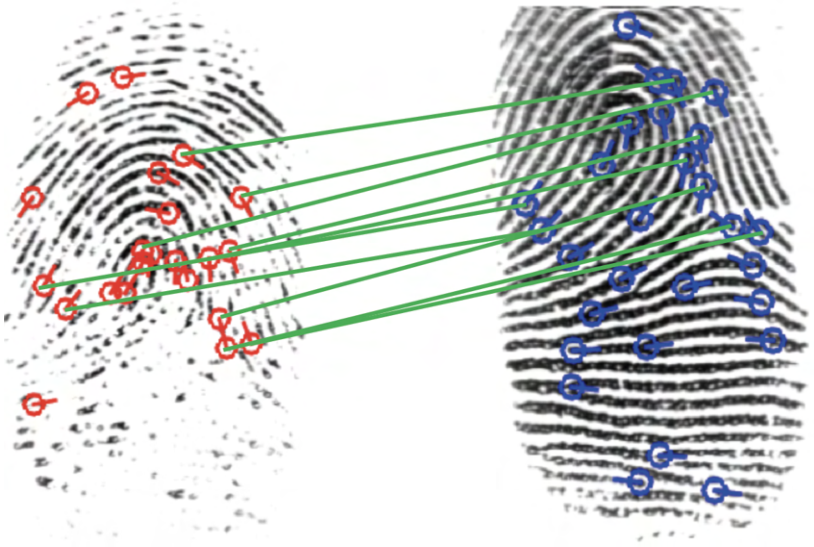


Figure 1.12 Example of minutiae match between two fingerprint impressions of the same finger. This example highlights the difficulty of minutiae matching given a poor quality enrollment image (left) which has many missing minutiae. Despite this difficulty, a COTS minutiae matcher is able to correctly match these two fingerprints with a score of 150, well above the score threshold of 69 @ $FAR = 0.01\%$. Fingerprints retrieved from the FVC 2004 DB1 A database [109]

.

must be adequately secured. Securing the template database in a manner that still allows the underlying fingerprint system to operate at high levels of accuracy remains a significant research challenge [87]. In Chapter 4 of this thesis, we show how a discriminative fixed-length representation can be learned, secured, and matched in the encrypted domain using fully homomorphic encryption.

1.3.4 Matching

The final step in the fingerprint recognition pipeline is that of matching, or comparing two templates. Typically the matcher will output a score s within some range (*e.g.* $s \in [0, 1]$). If the score is above a threshold t, then the decision is a match, otherwise, it is a non-match. The threshold t is selected in order to balance (according to the specific application, *e.g.* a mobile phone vs. government facility access) the false accepts and false rejects of the matcher.

The most common approach to fingerprint matching is that of minutiae matching. The goal of minutiae matching is to align two minutiae sets (frequently of variable, unknown length) and subsequently finding a maximum number of candidate corresponding minutiae (Fig. 1.12) [111]. Some of the challenges with minutiae matching include: (i) non-linear distortion between minutiae sets and (ii) spurious or missing minutiae [72]. The extraction of fixed-length representations from fingerprints, shown in Chapter 4, enables matching using simple distance metrics such as the cosine distance [48] which operates at orders of magnitude faster speeds than minutiae matching algorithms (a significant benefit for large scale fingerprint search applications).

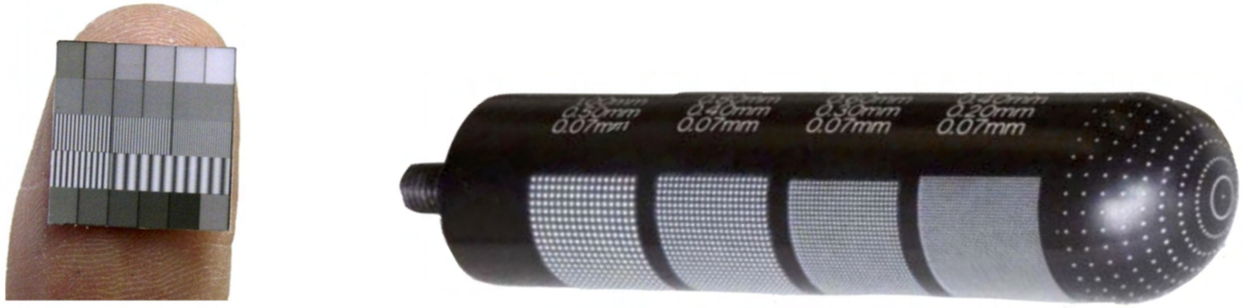
1.4 Remaining Challenges

Although fingerprint recognition systems have come a long way with respect to automation, accuracy, statistical understanding, and prevalence throughout society from the early days of Galton, Herschel, Henry, and other pioneers, current automated fingerprint recognition systems still face several limitations. The following section lists several of the limitations of prevailing fingerprint recognition systems. This list is by no means exhaustive, rather it is a specific list of problems which this thesis aims to address.

1.4.1 Fingerprint Reader Evaluations

Current automated fingerprint recognition systems make use of a plethora of different fingerprint readers, each employing different sensing technologies. Given that a particular recognition system may make use of multiple types of fingerprint readers, it is very important to have standardized evaluations (especially *interoperability* evaluations, where the fingerprint reader used during enrollment and authentication or search differ) for fingerprint readers to ensure that the recognition accuracy does not degrade as a result of poor fingerprint sensing.

Existing standards for fingerprint reader evaluations are primarily based upon the FBI *PIV* and *Appendix F standards* [55]. The *Appendix F* standard is comparatively stringent, requires pristine image capture, and is designed to facilitate evaluation of fingerprint readers used in person search scenarios (one to many comparisons). The *PIV* standard is a softer standard than *Appendix F* and is designed to evaluate fingerprint readers used in person verification scenarios (one to one comparison). Both of these standards use imaging targets that are fabricated by projecting a calibration pattern (e.g. sine gratings) onto a flat surface (Fig. 1.13). These targets are useful for structural (white-box) testing of fingerprint readers since they ensure that certain quantitative imaging thresholds are met by the fingerprint reader's sensing component, however, these targets have little resemblance to the human fingers that the readers will be exposed to in an operational setting. As such, controlled operational (black-box)[13] evaluations of fingerprint readers using the existing standards and targets are limited at best.



(a) Single Finger Reader Target

(b) 3D Contactless Reader Target

Figure 1.13 Examples of existing fingerprint reader calibration targets. These targets are useful for white-box testing fingerprint readers, ensuring that they meet certain quantitative imaging thresholds, however, they are very dissimilar from human fingers. As such they are not useful for realistic operational evaluations of fingerprint readers.

1.4.2 Fingerprint Presentation Attack Detection

An outstanding security flaw with fingerprint recognition systems is that of successful spoof attacks, or presentation attacks[14]. The most common type of presentation attack (referred to as

[13] White-box testing evaluates the internal sub-components of a system, whereas black-box testing focuses on testing the end-to-end system using system inputs and outputs [13].

[14] In ISO standard IEC 30107-1:2016(E), presentation attacks are defined as the *"presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system"* [81].

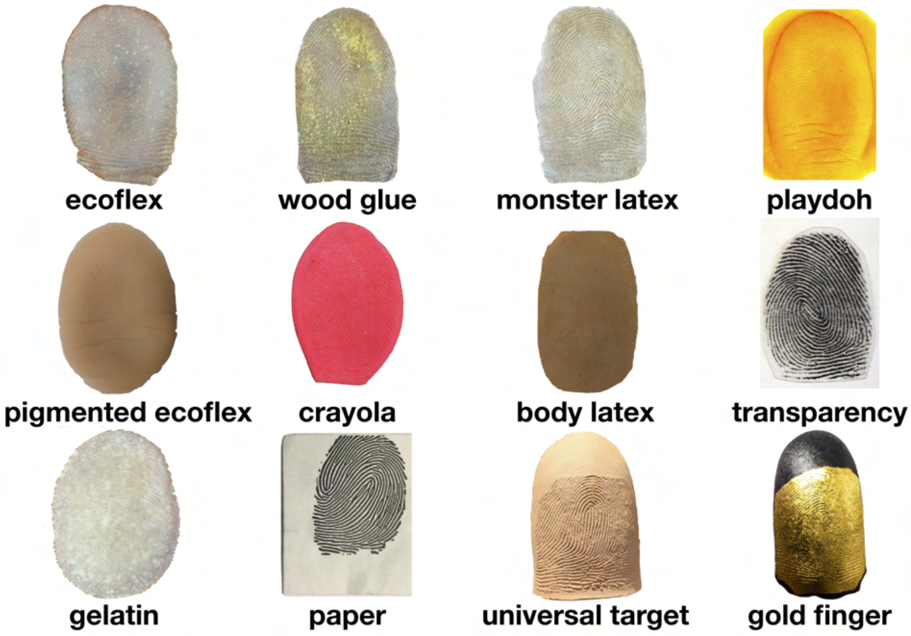
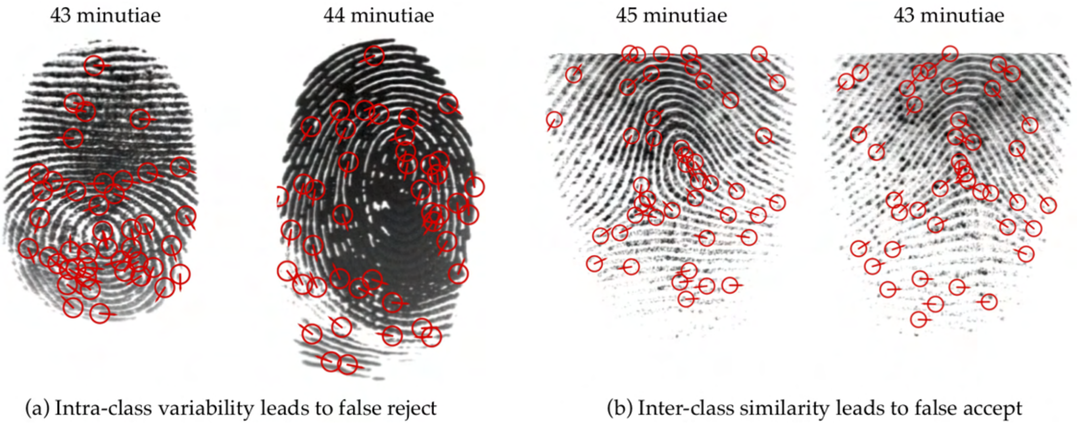


Figure 1.14 Examples of fingerprint spoofs made from different materials. The material variety demonstrates why it is difficult to develop a spoof detector which generalizes well across all material types.

spoofing) occurs when a *hacker* intentionally assumes the identity of unsuspecting individuals, called *victims* here, through stealing their fingerprints, fabricating spoofs (fake fingers made of common household materials like gelatin or wood glue) (Fig. 1.14) with the stolen fingerprints, and maliciously attacking fingerprint recognition systems with the spoofs into identifying the hacker as the victim[15] [113, 115, 187].

Over the last decade, a number of different approaches using either hardware or software have been proposed to automatically detect and flag fingerprint spoof attacks at the fingerprint reader to thwart these attacks [113]. However, most of these approaches do not meet adequate levels of accuracy for field deployment. Furthermore, many of the more novel learning based approaches fail when presented with spoofs fabricated from materials not seen during training of the spoof detector. In fact, several studies have reported up to a three-fold increase in error when testing spoof detectors on unknown material types [112, 167, 185]. As such, robust, generalizing fingerprint

[15] Presentation attacks can also occur when (i) two individuals are in collusion or (ii) an individual obfuscates his or her own fingerprints to avoid recognition [113]. However, in this thesis our specific aim is to stop fingerprint spoofing presentation attacks.



(a) Intra-class variability leads to false reject

(b) Inter-class similarity leads to false accept

Figure 1.15 Failures of a state-of-the-art COTS minutiae-based matcher (minutiae annotated with COTS). The genuine pair (two impressions from the same finger) in (a) was falsely rejected at 0.1% FAR (score of 9) due to heavy non-linear distortion and moist fingers. The imposter pair (impressions from two different fingers) in (b) was falsely accepted at 0.1% FAR (score of 38) due to the similar minutiae distribution in these two fingerprint images (the score threshold for COTS A @ FAR = 0.1% is 34). These slap fingerprint impressions come from public domain FVC 2004 DB1 A database [109]. The number of minutiae in FVC 2004 DB1 A images range from 11 to 87.

presentation attack detection remains a challenging, unsolved problem in fingerprint recognition systems.

1.4.3 Fixed-Length Fingerprint Representations

The most common type of fingerprint representation is that of an unordered, variable length minutiae set. Although AFIS based on minutiae representations (*i.e.* handcrafted features) have seen tremendous success over the years, they have several limitations.

- Minutiae-based representations are of variable length, since the number of extracted minutiae varies amongst different fingerprint images even of the same finger (Figure 1.15). Variations in the number of minutiae originate from a user's interaction with the fingerprint reader (placement position and applied pressure) and condition of the finger (dry, wet, cuts, bruises, etc.). This variation in the number of minutiae causes two main problems: (i) pairwise fingerprint comparison is computationally demanding and varies with number of minutiae and

(ii) matching in the encrypted domain, a necessity for user privacy protection, is computationally expensive, and results in loss of accuracy [87].

- In the context of global population registration, fingerprint recognition can be viewed as a **75 billion class problem** (≈ 7.5 billion living persons around the globe, assuming nearly all with 10 fingers) with large intra-class variability and large inter-class similarity. This necessitates extremely discriminative yet compact representations that are complementary and at least as discriminative as the traditional minutiae-based representation. For example, India's civil registration system, Aadhaar, now has a database of over 1.2 billion residents who are enrolled based on their 10 fingerprints, 2 irises, and face image [171].

- Reliable minutiae extraction in low quality fingerprints (due to noise, distortion, finger condition) is problematic, causing false rejects in the recognition system. See also NIST fingerprint evaluation FpVTE 2012 [178].

Given these limitations with the prevailing minutiae representation, it is desirable to extract discriminative *fixed-length* representations from fingerprints. Fixed-length representations can be compared extremely quickly using simple distance metrics, and can be matched securely in the encrypted domain using fully homomorphic encryption. Approaches in the literature attempting to extract fixed-length fingerprint representations fail to match the accuracy of traditional minutiae matchers [18, 89, 90, 159].

1.4.4 Infant Fingerprints

While fingerprint recognition systems are now being used around the world in a number of applications by billions of teenagers and adults, infants and young children remain excluded from these applications. Indeed, current automated fingerprint recognition systems fail to work on infants (0-12 months of age) and young children due to (i) small inter-ridge spacing (often less than 1 pixel in-between two ridges), (ii) non-linear distortion from soft infant skin, (iii) uncooperative subjects

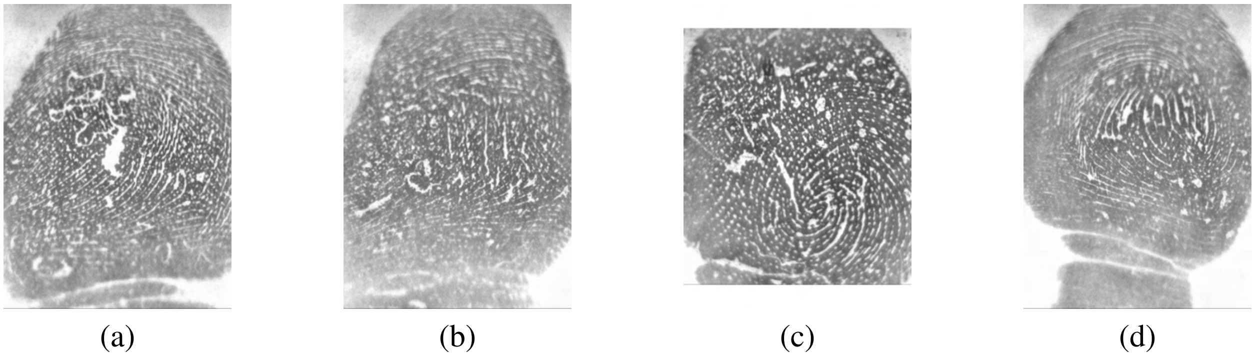

(a) (b) (c) (d)

Figure 1.16 Examples of low quality infant fingerprints. These examples demonstrate the difficulty in using automated fingerprint recognition systems for infant recognition. Note the small inter-ridge spacing, debris, motion blur, and moisture throughout the different impressions. These images were captured when the infant's were 2 months old.

causing motion blur, (iv) wet or dry fingers, and (v) debris, threads, or hairs from the infant's clothing or the mother's hair obfuscating the fingerprints (Fig. 1.16). This is alarming since national ID programs like India's Aadhaar now use fingerprints for providing government assistance and benefits distribution (Aadhaar starts enrollment at 5 years of age leaving younger ages excluded). Developing a fingerprint recognition system for infants could aid in establishing verifiable identity for these infants which could in turn be used for a myriad of applications such as robust vaccination tracking.

1.5 Contributions

This thesis aims to address each of the aforementioned limitations with state-of-the-art automated fingerprint recognition systems.

- **Universal 3D Wearable Fingerprint Targets**

  To enable robust, standardized fingerprint reader operational evaluations (especially interoperability), we present the fabrication of an interoperable 3D fingerprint target through a molding and casting process. We call our target the *universal fingerprint target*. Like previous fingerprint targets in [5], the universal fingerprint targets share a 3D geometry similar to a fingerprint surface, have mechanical properties similar to human skin, and are mapped with a fingerprint image, either real or synthetic. However, unlike previous fingerprint targets, the universal fingerprint targets are unique in that they incorporate the technically pertinent mechanical, optical, and electrical properties of the human skin within a single target, making it possible for the universal fingerprint targets to be imaged by all major fingerprint sensing technologies in use (capacitive, contact-optical, contactless-optical)[16]

- **RaspiReader: Open Source Fingerprint Reader**

  To address the security vulnerability of presentation attacks in fingerprint recognition systems, we open source a low cost, spoof resistant fingerprint reader, called RaspiReader. RaspiReader is an FTIR fingerprint reader customized with two cameras for image acquisition rather than a single camera. Use of two cameras enables robust fingerprint spoof detection, since we can extract features from two complementary, information rich images instead of processed grayscale images output by traditional COTS optical fingerprint readers. We demonstrate in our experimental results that RaspiReader enables a significant boost in spoof detection performance in comparison to COTS optical readers. We also show that RaspiReader is more generalizable to unseen materials than existing COTS readers.

- **Learning a Fixed-Length Fingerprint Representation**

  To overcome the limitations of prevailing minutiae-based matchers, we design a deep network embedded with fingerprint domain knowledge, called **DeepPrint**, to *learn* a fixed-length representation of 200 bytes which discriminates between fingerprint images from different fingers. While prevailing minutiae-matchers require expensive graph matching algorithms for fingerprint comparison, the 200 byte representations extracted by DeepPrint can be compared using simple distance metrics such as the cosine similarity, requiring only d multiplications and $d - 1$ additions, where d is the dimensionality of the representation

[16] We use the term universal to indicate that the targets can be imaged by all existing, major types of fingerprint readers (contact-optical, contactless-optical, capacitive, Ultrasound, and multi-spectral direct-view).

(for DeepPrint, $d = 192$)[17]. This fast comparison of DeepPrint representations is particularly useful for large-scale image search, when millions or even billions of comparisons must be made. Another significant advantage of this fixed-length representation is that it can be matched in the encrypted domain using fully homomorphic encryption [10, 14, 174, 175]. Finally, since DeepPrint is able to encode features that go beyond fingerprint minutiae, it is able to match poor quality fingerprints when reliable minutiae extraction is not possible. In short, DeepPrint enables faster, and more secure fingerprint matching than the prevailing minutiae representation with comparable levels of accuracy.

- **Infant-ID: Fingerprints for Global Good**

  To extend the use of fingerprint recognition for all ages, we have developed an end-to-end infant fingerprint recognition system. We have prototyped a high-resolution (1,900 ppi) reader designed for infants. Using this reader, we have captured a longitudinal dataset (data acquired over a time lapse of 1 year, in 4 sessions) of 315 infants from a rural clinic in Agra, India. To match these high-resolution infant fingerprints, we have developed a high-resolution infant fingerprint matcher. Finally, we have demonstrated that by using our infant fingerprint reader and matcher, we are able to enroll infants at 2-months of age, and recognize them a full year later. This allows for our infant matcher to be used to alleviate infant suffering around the world by providing every infant a digital and verifiable identity which could be used for vaccination tracking, food distribution, and government assistance later in life.

[17] The DeepPrint representation is originally 768 bytes (192 features and 4 bytes per float value). We compress the 768 bytes to 200 by scaling the floats to integer values between [0,255] and saving the two compression parameters with the features. This loss in precision (which saves significant disk storage space) very minimally effects matching accuracy.

Chapter 2

Universal 3D Wearable Fingerprint Targets: Advancing Fingerprint Reader Evaluations

In this chapter, we propose a manufacturing process to create Universal 3D Wearable Fingerprint Targets for repeatable and controlled fingerprint reader evaluations [43]. The Universal Targets are designed to be imaged across a variety of fingerprint sensing technologies including capacitive, contact-optical, and contactless-optical and as such, they are particularly well suited for fingerprint reader interoperability studies. Fingerprint reader interoperability refers to how robust fingerprint recognition systems are to variations in the images acquired by different types of fingerprint readers. To build universal 3D fingerprint targets, we adopt a molding and casting framework consisting of (i) digital mapping of fingerprint images to a negative mold, (ii) CAD modeling a scaffolding system to hold the negative mold, (iii) fabricating the mold and scaffolding system with a high resolution 3D printer, (iv) producing or mixing a material with similar electrical, optical, and mechanical properties to that of the human finger, and (v) fabricating a 3D fingerprint target using controlled casting. Our experiments conducted with PIV and Appendix F certified optical (contact and contactless) and capacitive fingerprint readers demonstrate the usefulness of universal 3D fingerprint targets for controlled and repeatable fingerprint reader evaluations and also fingerprint reader interoperability studies.

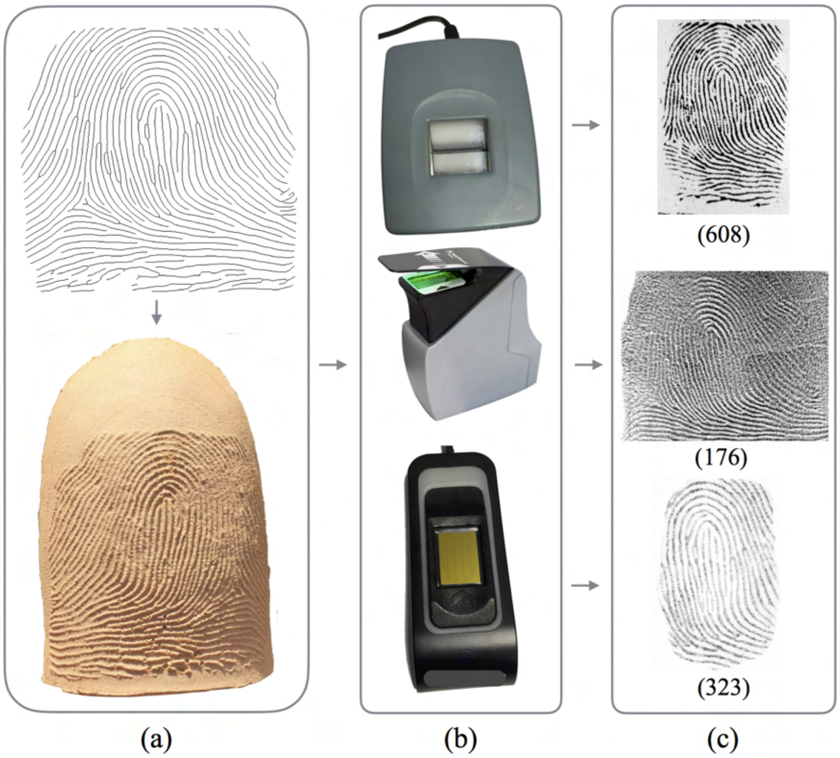


Figure 2.1 A Universal 3D Fingerprint Target fabricated in (a) can be imaged by a variety of popular fingerprint readers (contact-optical, contactless-optical, and capacitive) shown in (b). The sensed images of the 3D fingerprint target in (a) are shown in (c). This demonstrates that our targets are appropriate for fingerprint reader interoperability evaluation studies. Similarity scores for each sensed fingerprint image (with the 2D mapped target image) are displayed below each fingerprint image in (c). Verifinger 6.3 SDK was used for generating similarity scores. The score threshold at 0.01 % FAR is 33.

2.1 Introduction

The current *PIV* and *Appendix F* evaluation standards for fingerprint readers use simple calibration patterns to ensure certain imaging quality thresholds are met. However, because these targets are dissimilar (optically, electrically, and mechanically) from the real human fingers the readers will be imaging in an operational scenario, current evaluation standards are limited at best.

To address the challenges of robust operational evaluation inherent to imaging devices, the medical imaging community has developed 3D targets (phantoms) as evaluation specimens. Phantoms are useful for evaluating a variety of medical imaging devices in areas such as radiography, tomography, and ultrasonic imaging [116] [135]. Use of live subjects for repeated evaluation of medical devices is impractical because of the health hazards and monetary costs involved. How-

Figure 2.2 Example phantom of a human hand [116] used in the medical domain.

Table 2.1 Properties of the Human Finger [42, 51, 52]

Shore A Hardness	Tensile Strength (MPa)	Elongation at Break (%)	Electrical Resistivity (Ω-cm)
20-41	5-30	35-115	$2.5 * 10^2 - 8 * 10^6$

ever, realistic 3D phantoms (Fig 2.2) make accurate operational evaluation of these devices possible. Proper operational evaluation of fingerprint readers can only be accomplished, in a similar manner, by using 3D fingerprint targets (phantoms) with similar characteristics to the human finger.

2.1.1 3D Fingerprint Targets

Some research has been conducted developing 3D targets towards achieving the aforementioned goal. In 2011, Orandi *et. al* developed 3D cylindrical metal targets mapped with 2D calibration patterns for contactless fingerprint readers [134]. However, because these targets are rigid and completely dissimilar in mechanical, optical, and capacitive properties to the human finger, they can not be used by contact-based fingerprint readers. More recently, in 2016, Arora *et. al* produced high fidelity 3D fingerprint targets using a high resolution, state-of-the-art 3D printer [5] [4] [6].

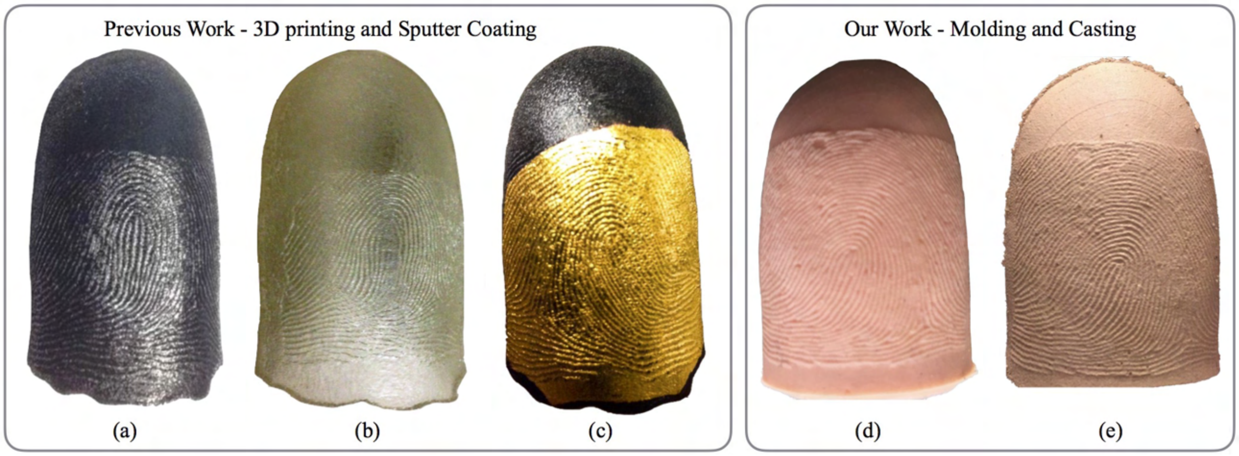

Previous Work - 3D printing and Sputter Coating
Our Work - Molding and Casting
(a)
(b)
(c)
(d)
(e)

Figure 2.3 High fidelity, wearable, 3D fingerprint targets. (a) 3D fingerprint target printed using TangoBlackPlus FLX980 [5], (b) 3D fingerprint target printed using TangoPlus FLX 930 [4], (c) 3D fingerprint target printed using TangoBlackPlus FLX980 and then sputter coated with 30 nm titanium + 300 nm of gold [6], (d) our casted 3D fingerprint target using a mixture of PDMS (Polydimethylsiloxane) and Pantone 488C color pigment [157] [40], and (e) our casted universal 3D fingerprint target using a mixture of conductive PDMS, silicone thinner, and Pantone 488C color pigment [157] [154] [158]. 3D targets in (a), (b), and (c) were printed on a high resolution 3D printer (Stratasys Objet350 Connex).

These targets were a big step forward in the direction of realistic operational fingerprint reader evaluation because the targets employed a 3D geometry similar to the human finger, they were fabricated using materials with similar mechanical properties as human skin, they were mapped with real fingerprint images, and they could be worn on a human finger. However, due to the limited number of materials that can be used in 3D printers, the polymers used for printing (i) did not have the same nominal electrical conductivity of human skin and (ii) did not have the spectral reflectance of human skin. As a result, multiple types of targets (Figs. 2.3 (a), (b), (c)) were fabricated for different types of fingerprint readers (capacitive, contact-optical, and contactless-optical) [5] [4] [6]. These individual targets worked for evaluating the type of reader for which they were designed, however, they were not interoperable. That is, a target fabricated for one type of fingerprint reader (e.g. capacitive) would not work on a different type of fingerprint reader (e.g. optical). Because multiple types of targets were needed for evaluating different types of readers, performing a standardized interoperability evaluation of fingerprint reader technologies was not possible with these 3D printed targets.

Table 2.2 Properties of published 3D Printed Targets [4, 5, 9]

Specimen Material	Shore A Hardness	Tensile Strength (MPa)	Elongation at Break (%)	Color	Electrical Resistivity (Ω-cm)	Cost (USD)
TangoBlackPlus FLX980 (Fig. 2.3 (a)) [5] [161]	26-28	0.8-1.5	170-220	Black	Insulator	$10.00
TangoPlus FLX930 (Fig. 2.3 (b)) [4] [161]	26-28	0.8-1.5	170-220	Translucent	Insulator	$10.00
TangoBlackPlus FLX980, Ti-Au coating (Fig. 2.3 (c)) [6] [161] [9]	26-28	0.8-1.5	170-220	Gold	$2.4 * 10^{-5}$	$12.00

2.1.2 Fingerprint Reader Interoperability

Past studies on fingerprint reader interoperability have shown that when different fingerprint readers were used for enrollment and identification (or verification), some loss in recognition accuracy ensued [144] [145] [120]. However, all of these studies were performed on data acquired from live human subjects [91]. As such, variations (finger pressure and orientation; conditions of the finger, *e.g.* wet or dry) between impressions on the different readers could account for some of the error observed. We posit that in order to truly quantify the effects of interoperability, an interoperable fingerprint target would need to be mounted to a robot gripper and imaged on different readers at the same pressure and orientation.

As noted in [119], continued advances in distributed computing have enabled less monolithic fingerprint recognition systems. This advent of larger, more distributed systems (*e.g.* the Aadhaar system) drastically increases the likelihood that the fingerprint reader used to enroll a user's fingerprint image at one location will not be the same reader (or model of reader) used later to identify or verify the same individual at another location. Furthermore, even if the same reader

Table 2.3 Properties of our 3D Casted Targets

Specimen Material	Shore A Hardness	Tensile Strength (MPa)	Elongation at Break (%)	Color	Electrical Resistivity (Ω-cm)	Cost (USD)
PDMS, Pantone 488C (Fig. 2.3 (d)) [40]	43	6.7	120	PMS 488C	Insulator	\$0.86
Conductive PDMS, Thinner, Pantone 488C (Fig. 2.3 (e)) [157] [154] [158]	38.5	2.0	80	Tan + PMS 488C	$9.8 * 10^{-1}$ †	\$10.00

† Although the resistivity of the target differs from human skin, the resistivity value is sufficient for image capture by capacitive readers.

is used for both enrollment and identification, advances in sensing technology could eventually require replacement of the reader being used. As mentioned in [120], the cost to an institution needing to re-enroll its entire database of users on a new reader could be monumental. Both of these situations underscore the need to know and quantify fingerprint reader interoperability. If fingerprint recognition systems are to continue to become more distributed, then the performance change associated with interoperability must be objectively known and quantified. Doing so will benefit system users, reader manufacturers, system developers, and the institutions deploying the system.

2.1.3 Universal 3D Fingerprint Targets

To enable robust, standardized fingerprint reader interoperability evaluations, we present the fabrication of an interoperable 3D fingerprint target (Fig. 2.1) through a molding and casting process (Fig. 2.4). We call our target the *universal fingerprint target* (Fig. 2.3 (e)). Like previous fingerprint targets in [5], the universal fingerprint targets share a 3D geometry similar to a fingerprint surface, have mechanical properties similar to human skin, and are mapped with a fingerprint image, either real or synthetic. However, unlike previous fingerprint targets, the universal fingerprint

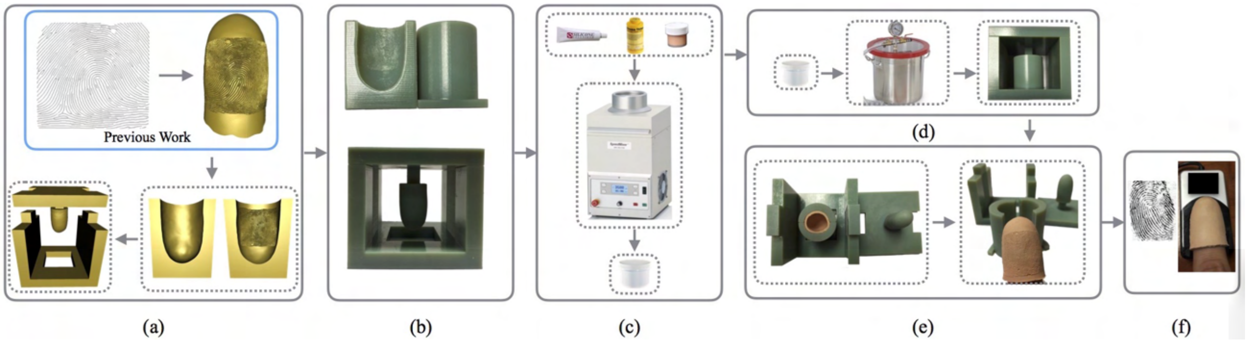


Figure 2.4 System block diagram of the proposed molding and casting process for making 3D targets. (a) A 3D negative mold (of a 2D fingerprint image) and a supporting scaffolding system (necessary for making the fingerprint target wearable) are electronically fabricated; (b) 3D electronic models are manufactured by 3D printing and chemical cleaning; (c) conductive silicone, silicone thinner, and human colored dye are mechanically mixed to produce a casting material with similar conductive, mechanical, and optical properties to the human skin; (d) the material fabricated in (c) is cast into the mold and scaffolding system; (d) vacuum degassing ensures that air bubbles are removed from the casted material; (e) wearable fingerprint targets are extracted 72 hours after pouring the casting material; (f) the wearable, 3D fingerprint target is used for fingerprint reader evaluations.

targets are unique in that they incorporate the technically pertinent mechanical, optical, and electrical properties of the human skin within a single target (Tables 2.1, 2.2, and 2.3), making it possible for the universal fingerprint targets to be imaged by all major fingerprint sensing technologies in use (capacitive, contact-optical, contactless-optical)[1]. The universal fingerprint targets enable and facilitate, for the first time, a standardized assessment of fingerprint reader interoperability. The universal fingerprint targets also enable controlled data collection useful for fingerprint distortion modeling.

More concisely, the contributions of this chapter are:

- A controlled, repeatable process for creating fingerprint target molds, and fabricating high quality finger castings. Unlike previous works [5], this casting fabrication process is not

[1] We use the term universal to indicate that the targets can be imaged by all existing, major types of fingerprint readers (contact-optical, contactless-optical, and capacitive). Since the submission of this manuscript, we have also verified that the targets can be imaged by Optical Coherence Tomography (OCT), ultrasound, and multispectral fingerprint readers. If new sensing technologies emerge requiring additional properties in casted targets, our flexible casting process allows for concocting a new material with the necessary properties. So, our process can be easily extended to manufacture targets for new fingerprint sensing technologies that may emerge.

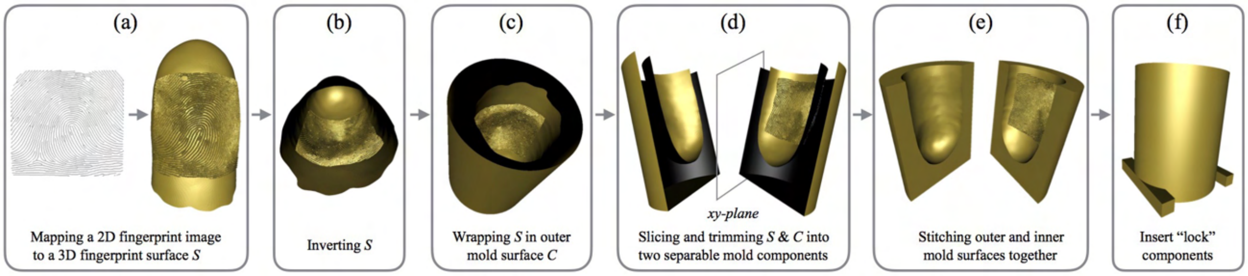

(a)
Mapping a 2D fingerprint image to a 3D fingerprint surface S
(b)
Inverting S
(c)
Wrapping S in outer mold surface C
(d)
xy-plane
Slicing and trimming S & C into two separable mold components
(e)
Stitching outer and inner mold surfaces together
(f)
Insert "lock" components

Figure 2.5 Process flow for fabricating electronic 3D fingerprint mold, M

restricted to a small number of materials. Additionally, it is not cost prohibitive as it is based on a potentially high-throughout casting process.

- Fabricating high fidelity universal 3D fingerprint targets with similar mechanical, optical, and electrical properties to the human skin. Previous targets did not simultaneously possess both the optical and electrical properties of human skin within a single target.

- Fingerprint image capture, using the same 3D target, from optical readers (contact and contactless) and capacitive readers. Our universal fingerprint targets enable standardized interoperability data collection for the first time ever.

- Experimental evaluations, using the universal 3D fingerprint targets and three different types of commercial off-the-shelf (COTS) fingerprint readers[2] (contact-optical, contactless-optical, and capacitive). Our results quantify the loss in fingerprint recognition accuracy when different readers are used for enrollment and identification (or verification). These findings validate the use of our universal 3D fingerprint target for further fingerprint reader interoperability studies.

[2] Because of our Non-Disclosure Agreement with the vendors, we cannot provide the make and model of the readers used in our experiments.

2.2 Mold & Scaffold Fabrication

To fabricate a fingerprint target T, we begin by electronically modeling (and subsequently manufacturing) a fingerprint mold M and scaffolding framework F.

2.2.1 Mold Fabrication

First, a negative[3] fingerprint mold is electronically designed (Fig. 2.5), 3D printed, and chemically cleaned. This process is further broken down in the following steps.

i) Inner Mold Surface - Using techniques similar to [5], a 2D fingerprint image is mapped onto a smooth 3D finger surface mesh S in a manner that retains the topology inherent to the 2D image (Fig. 2.5 (a)). More formally, let S be a mesh of triangular faces $F = \begin{bmatrix} f_1, & f_2, & f_3, & ..., & f_n \end{bmatrix}$, and 3-dimensional vertices $V = \begin{bmatrix} v_1, & v_2, & v_3, & ..., & v_c \end{bmatrix}$. Each face in F is explicitly defined as an ordered list of 3 vertices from V, *e.g.* $f_1 = \begin{bmatrix} v_i, & v_j, & v_k \end{bmatrix}$. Additionally, every face in F contains a normal vector which is implicitly encoded by the order of the 3 vertices used to define the face. In particular, the direction of the normal vector is determined by taking the cross product of the vectors formed with respect to the order of the face's three vertices. For example, the normal vector for face f_1 is $f_{1,normal} = a \times b$, where a is a vector having tail at v_i and head at v_j, while b is a vector having tail at v_j and head at v_k.

Because the end goal of the electronic modeling of M is to produce a negative mold, the mapped surface S must be inverted by flipping all the faces of S (Fig. 2.5 (b)). For every face, this flipping is attained by reversing the order of its three vertices - and consequently the implicitly encoded direction of its normal vector. For example, by changing $f_1 = \begin{bmatrix} v_i, & v_j, & v_k \end{bmatrix}$ to $\hat{f}_1 = \begin{bmatrix} v_k, & v_j, & v_i \end{bmatrix}$, the normal vector $\hat{f}_{1,normal}$ computed by $a \times b$ is reversed in direction, since a is now a vector having tail at v_k and head at v_j, while b is a vector having tail at v_j and head at v_i.

ii) Outer Mold Surface - After iteratively inverting all n $\begin{bmatrix} f_1, & f_2, & f_3, & ..., & f_n \end{bmatrix}$ faces, the next step in generating mold M is to imprint the fingerprint surface S inside of an open ended

[3] In molding and casting, positive sculptures are produced from their negative mold.

cylindrical surface C (Fig. 2.5 (c)). Surface C acts as the exterior of the final mold M. As such, dimensions for C are determined empirically so as to provide strength and durability to the mold and to prevent usage of excess material. Our experiments show that setting the height of C to $C_{height} = 1.25 * S_{height}$ balances the need for structural support and minimizes material cost for casted targets (here S_{height} is the height of the fingerprint surface S). The diameter of the mold (C_{dia}) is fixed at 34 mm. While C_{dia} could have been dynamically chosen based upon the diameter of S (S_{dia}), we chose a fixed value so that all the molds we print could fit within a single scaffolding framework F. We chose 34 mm as a static diameter value, since the 95th percentile of the widest adult finger (the thumb) is 26 mm to 27 mm [62]. As such, the minimum thickness (t_{min}) of our mold is computed as $t_{min} = 1/2 * (34 - 27)mm = 3.5$ mm. We empirically validated that a mold thickness of $t_{min} \geq 3.5$ mm provides the durability needed for our casting process.

iii) Split Mold - With the inner and outer surface of the mold in place, we continue the fabrication process by simultaneously splitting C and S along the *xy-plane* into C_{above}, S_{above}, C_{below}, and S_{below}. Splitting the mold into two semi-cylindrical components will facilitate the extraction of the final fingerprint castings T (from the mold). C_{above}, S_{above}, C_{below}, and S_{below} are further post processed by adding new faces and vertices such that all four surfaces lie flat on the *xy-plane*. Figure 2.5 (d) illustrates the sliced, trimmed, and post processed components C_{below}, C_{above}, S_{below}, and S_{above}.

iv) Stitching and Printing - Finally, the individual surfaces C_{below} and S_{below} and C_{above} and S_{above} are stitched together into two three-dimensional, semi-cylindrical mold halves by adding triangular faces around the periphery of the respective surfaces. Upon completion of this stitching, a high fidelity fingerprint mold M has been electronically fabricated (Fig. 2.5 (e)).

To minimize the variability of fingerprint targets during consecutive castings, two *"lock"* components are attached to the bottom of C (Fig. 2.5 (f)). These lock pieces, having length equal to 34 mm (C_{dia}) will prevent C from rotating inside of the scaffolding framework F.

At this point, M is physically realized by using a high resolution, state-of-the-art 3D printer that has the ability to print in slices as small as 16 microns [162]. A printer with such fine resolution is necessary to capture the minute details of the mapped fingerprint onto M[4]. As in [5], the mold is printed in 30 micron layers as this captures the necessary detail of the mapped fingerprints, while simultaneously decreasing the print time of M from 8 hours to 4 hours [5]. At the conclusion of printing, the mold is soaked in 2M NaOH[5] for about 4 hours to dissolve away the support material from the printed mold in a manner that does not damage the fingerprint ridges. After chemical cleaning a high fidelity fingerprint mold is ready for casting fingerprint targets (Fig. 2.6).

The resultant mold will only produce a solid casting, since casting material will fill the entire mold cavity. To make the cast wearable (*e.g.* mounting to a robotic gripper) or manual evaluation (*e.g.* human placement of the target) a *"scaffolding framework"* F is fabricated, which, when used in conjunction with M, creates a wearable 3D target T (Fig. 2.7). The process for generating F is further expounded upon below.

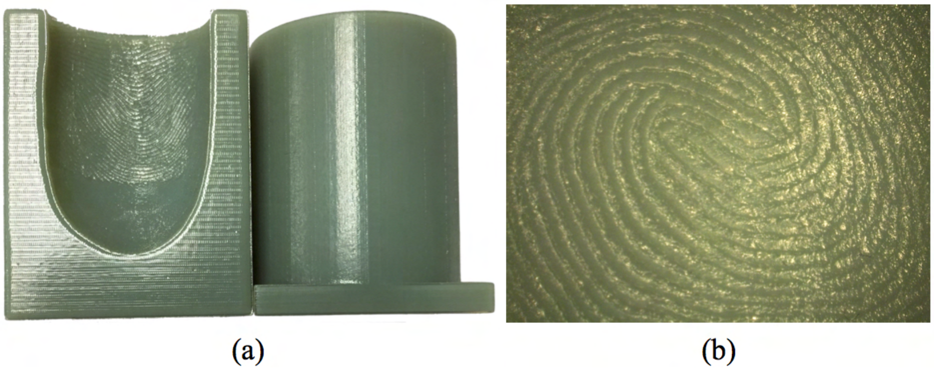


Figure 2.6 (a) High fidelity 3D printed fingerprint mold M. (b) View of fingerprint engraving on M at 20X magnification. The magnified image in (b) shows that all the friction ridge patterns are clearly present in the mold M. These friction ridge patterns are inverted, since negative molds are necessary to produce positive fingerprint targets (Fig 7 (c)).

[4] We also experimented with low resolution printers, however, the resolution was insufficient to cleanly separate the ridges and valleys of a fingerprint pattern.

[5] NaOH (Sodium Hydroxide) is a basic (alkaline) solution that cleans the residual printing support material away from the mold.

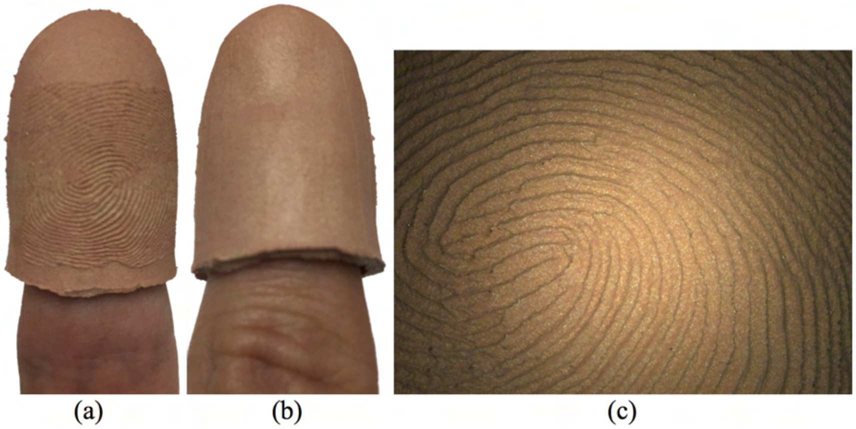


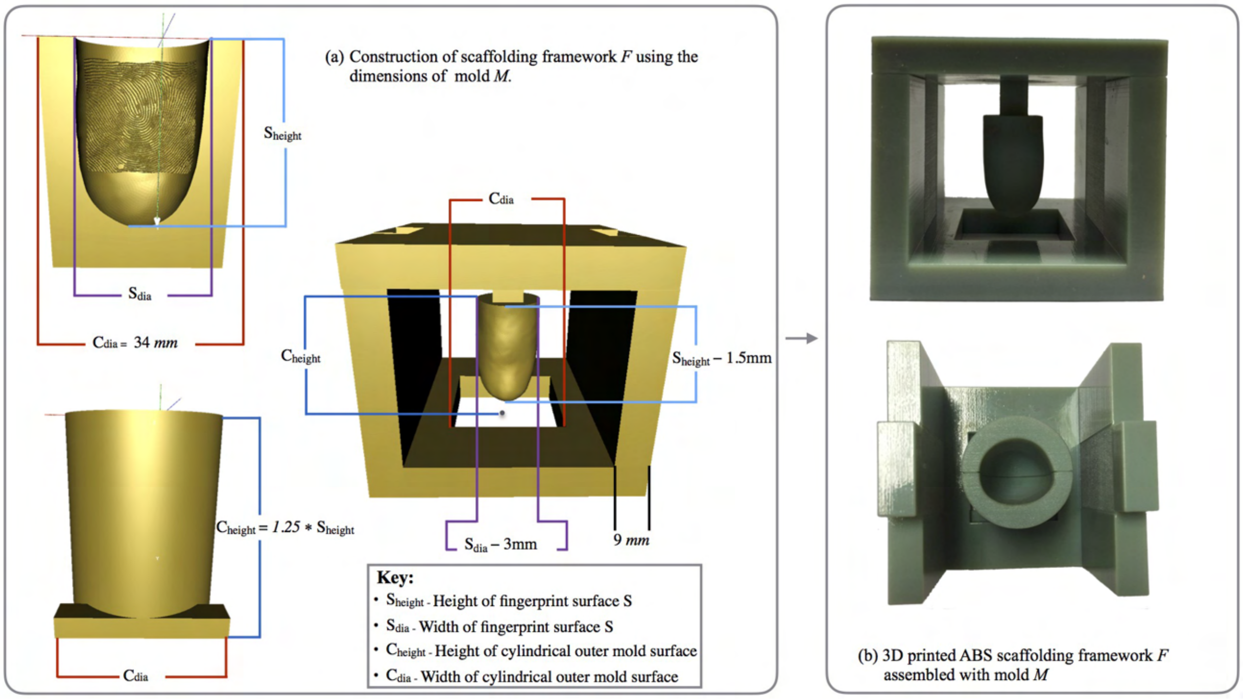
Figure 2.7 3D wearable Universal Fingerprint Target (a) front view, (b) rear view, and (c) view of the Universal Fingerprint Target ridges at 20X magnification.

2.2.2 Scaffolding Fabrication

To create a wearable fingerprint casting, a hollow, appropriately shaped void must be cured into the casted material as it resides in M. This void enables wearability as it creates the space where an end user's finger (or robotic attachment) would reside during evaluation.

We build upon the above idea by developing (based upon the dimensions M) a scaffolding framework F used to insert a fingerprint surface S' (with diameter slightly smaller than S_{dia}) into M during successive fingerprint target casts (Fig. 2.8 (a)). In doing so, we ensure that when casting material is injected into the mold, the space between S and S' will be filled to form a wearable fingerprint target T.

The scaffolding F consists of several components: a base platform that holds the mold M in place, two sides extending beyond the top of M, and a top piece from which the fingerprint surface S' is suspended. Aside from S', all of these pieces are generated by creating a simple cuboid shape and applying affine transformations until the component is of the correct size and in the correct position. The thickness of scaffolding walls is chosen to be 9 mm, which provides the structural robustness and durability needed for repeated castings of fingerprint targets. In addition, a concentric rectangular prism is cut from the inside of the base component. The length and width of this rectangular prism share the same dimension (C_{dia}) as the diameter of M. This ensures that



(a) Construction of scaffolding framework F using the dimensions of mold M.

(b) 3D printed ABS scaffolding framework F assembled with mold M

Sheight
Sdia
Cdia = 34 mm
Cdia
Cheight
Sheight − 1.5mm
Cheight = 1.25 * Sheight
Sdia − 3mm
9 mm
Cdia
Key:
• Sheight - Height of fingerprint surface S
• Sdia - Width of fingerprint surface S
• Cheight - Height of cylindrical outer mold surface
• Cdia - Width of cylindrical outer mold surface

Figure 2.8 Fabricating scaffolding F using the dimensions of the mold, M. (a) scaffolding framework F is electronically modeled; (b) the electronic scaffolding system is physically generated in acrylonitrile butadiene styrene (ABS) using a high resolution 3D printer. Using F in conjunction with M, 3D wearable fingerprint targets T are repeatably produced.

M will attach securely into the base unit, thus controlling the thickness of the casted targets. Since the diameter of M is fixed (based upon the 95th percentile of the human finger width at 34 mm), any mold can be attached interchangeably into a single scaffolding system.

Given that S' is a fingerprint surface with a diameter smaller than S, we can derive S' from the same scanned fingerprint surface that we originally used to generate S. That is, given a smooth scanned 3D fingerprint surface S_{smooth}, we can generate S' by shrinking S_{smooth} along the direction of its normals by 1.5 mm. More formally, if $v_1 = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}$ is a vertex of S_{smooth} and $n_1 = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}$ is the corresponding normal vector to v_1, then generating the new vertex

$v' = \begin{bmatrix} v'_x & v'_y & v'_z \end{bmatrix}$ for S' is computed as:

$$v' = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} - \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \times 1.5 \tag{2.2.1}$$

After all vertices of S_{smooth} have been iteratively shrunken along the direction of their corresponding normals, the top of S' is stitched shut using a triangle fan[6].

As with M, the electronic model of F is 3D printed using the same high resolution printer and parameters (Fig. 2.8 (b)). F is also cleaned with 2M NaOH solution to remove residual printing support material. Although F does not have the minute detail that M does, high resolution printing is still needed for printing F so that registration between F and M is consistent and reproducible. This ensures the high fidelity of the casted targets is preserved.

Upon completed fabrication of both M and F, we have tools for repeatably casting high fidelity, 3D wearable fingerprint targets T.

2.3 Casting

With tools developed for molding and casting in place, we next discuss the characteristics necessary (to emulate human skin) in the casting material for the 3D Universal Fingerprint Target. Additionally, we prescribe a process for concocting a material consisting of these characteristics and subsequently casting the material into a fingerprint mold and scaffolding system.

2.3.1 Material Requirements

Our material selection needs to carefully consider the optical, electrical, and mechanical properties inherent to the human finger.

[6] A triangle fan is a circular mesh surface, formed by placing a center vertex and filling in the circle with triangles that all share the center vertex.

- **Optical Property:** Optical readers rely on proper reflectance and refraction of light rays on the human finger surface to detect a fingerprint. Therefore, the optical properties of the targets must be similar to that of human skin to be accurately sensed by optical readers. Materials that are black will improperly absorb all light rays and materials of high reflectivity will improperly scatter all light rays, in both cases preventing targets of these materials from being imaged by many optical readers.

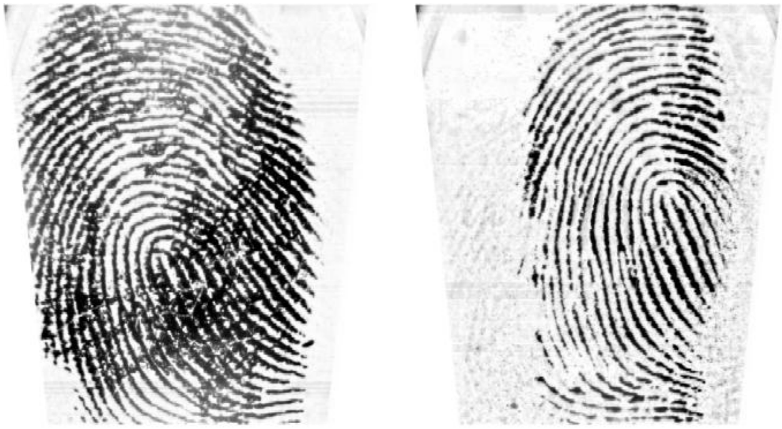
- **Electrical Property:** In addition to the color attribute, the targets must also be inherently conductive to act as a conductive plate and create capacitive differences between ridges and valleys on the cells within the semiconductor chips on capacitive sensors.

- **Mechanical Property:** Finally, the mechanical properties of the target material must lie within the range inherent to the human epidermis to ensure high quality fingerprint target image acquisition. Materials that deviate from the elasticity of the human epidermis could negatively impact the target in several ways. If the elasticity is too large, the minute details of the minutia will be lost as the target is compressed against the sensor and the ridges collapse under the force being exerted (Fig. 2.9 (a)). If, on the other hand, the elasticity is too small, or the hardness is too great, the fingerprint target will not flatten around the sensor platen, resulting in only partial print images of the fingerprint surface (Fig. 2.9 (b)).

2.3.2 Material Fabrication and Casting Procedure

To achieve the electrical, mechanical, and optical criteria necessary for the universal fingerprint target, electrically conductive silicone (SS-27S) [154] is sheer mixed [58] with silicone thinner [158] (at 4 % by mass), and a flesh-toned pigment [157] (at 3 % by mass)[7]. This casting material mixture is transferred to the mold from a disposable pipet. Prior to the transfer, both the mold and

[7] A simpler casting material - useful for interoperability assessment of contact and contactless optical readers - can be fabricated by mixing (with the FlakTek [58]) pure PDMS and PMS 488C pigment. These targets are not conductive, and are therefore unusable for capacitive reader evaluation, but they are optically and mechanically similar to the human finger and are cheaper to manufacture (Fig. 2.3 (d)) (Tables 2.1 and 2.3).



(a) 900 % elongation at break (b) Shore A 50

Figure 2.9 Fingerprint impressions captured from targets lacking proper mechanical characteristics. Notice (a) the presence of aberrations resulting from excessive elasticity in the target and (b) partial impression due to excessive hardness of the target.

scaffolding system are spray coated with silicone release agent [156]. After the material transfer, vacuum degassing at 98 kPa (0.97 atm) removes all air bubbles from the material. Finally, the mold and scaffolding system are assembled and left to cure (Fig. 2.4 (d)). After 72 hours, a high fidelity, 3D wearable universal fingerprint target, T, can be carefully extracted from the fingerprint mold and scaffolding system (Fig. 2.4 (e)).

2.3.3 Material Characterization

To verify the optical similarity of our fabricated material to human skin, we obtain a spectrogram [137] of the Universal Fingerprint Target material and compare it to a range of human skin spectrograms obtained by NIST [30] from 51 human subjects (Fig. 2.10). From this spectrogram, it can be seen that the spectral reflectance of the Universal Fingerprint Target material lies within the range of human skin for almost the entire visible spectrum (400 nm - 700 nm). At approximately 625 nm to 700 nm the Universal Fingerprint Target material does deviate from the range of human skin (.05 - .1 reflectance factor). Based on the NIST report, spectral reflectance varies significantly even across multiple readings of the same subject. Furthermore, only 51 subjects were evaluated to establish the range shown in Figure 2.10. Therefore, it is entirely possible that the Universal

Fingerprint Target material does lie within the spectral reflectance of human skin from 625 nm to 700 nm as well, given a larger number of subjects.

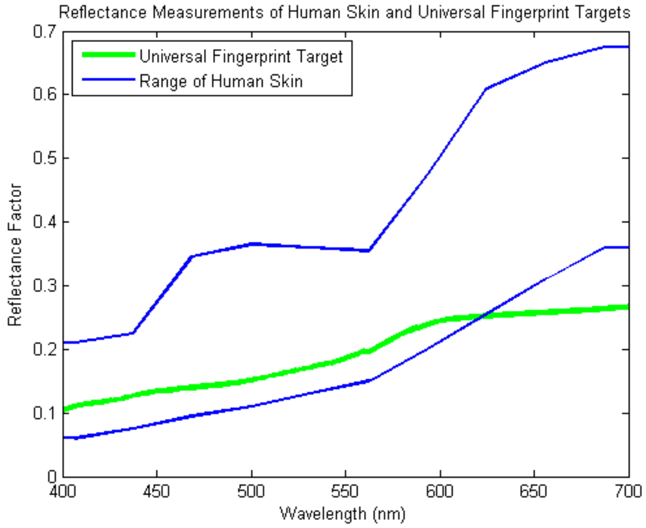


Figure 2.10 Comparison of the Universal Fingerprint Target material spectrogram to a range of spectrograms obtained by NIST from 51 human subjects. We plotted the range using estimated data points from the figure in [30].

In addition to verifying optical similarity of the material to human skin, we also verify that the material is electrically conductive by obtaining a resistivity reading (using [61]) from 4 square samples of the material. The average resistivity of the 4 samples is reported in Table 2.3.

Finally, the mechanical properties of the material are computed (using the data-sheets in [154] and [158]) and reported in Table 2.3. From the mechanical values reported in Tables 2.1 and 2.3, it can be seen that the chosen material is indeed within the range of the mechanical properties of human skin.

2.4 Target Fidelity and Reproducibility

To establish the universal fingerprint targets as standard evaluation artifacts, we must show that the proposed fabrication process (i) is of high fidelity and (ii) is reproducible. Both of these criterion are verified in the following subsections.

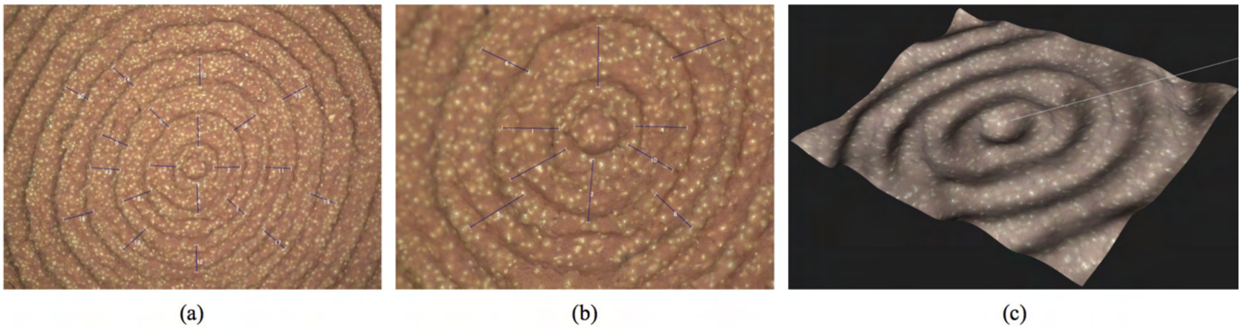


Figure 2.11 Images of the universal fingerprint target (mapped with circular sine gratings) captured using a Keyence optical microscope [98]. Point-to-point ridge distances are measured. (a) Image at 50X magnification and annotated with 20 point-to-point distances. (b) Image at 100X magnification and annotated with 10 point-to-point distances. (c) 3-D image generated by the microscope which qualitatively illustrates the uniformity in ridge height of the circular gratings on the universal fingerprint target. The granular texture in (a), (b), and (c) is evidence of the aluminum coated silver particles mixed into the universal fingerprint target which allows the target to be imaged by capacitive fingerprint readers.

2.4.1 Fidelity

A 3D universal fingerprint target is of high fidelity if its 3D ridges retain the topology inherent to the original 2D image it was fabricated from. We posit that fidelity of universal fingerprint targets can be objectively determined and compensated by quantifying the errors (as a deviation of the 3D target topology from the topology of the 2D mapping pattern) at each step in the fabrication process (Fig. 2.4) and accounting for these errors during fabrication.

(i) Error in Electronic Modeling of Fingerprint Mold - Arora *et al.* [5] showed that the projection algorithm used to map the 2D fingerprint image to a 3D finger surface results in a 5.8 % decrease in point-to-point distances inherent to the original 2D fingerprint image. Because the electronic fabrication of the fingerprint mold (Fig. 2.4 (a)) uses the same 2D to 3D projection algorithm as [5], the same error will be encountered in our universal fingerprint target fabrication process.

(ii) Error in 3D printing - Arora *et al.* [5] also observed an 11.42 % decrease in point-to-point distances (inherent to the original 2D fingerprint image) when fabricating the physical 3D target on a high resolution 3D printer. Since printing the fingerprint mold in (Fig. 2.4 (b)) was performed

using the same printer as [5], the universal fingerprint target fabrication process will encounter the same error.

While the errors introduced in both electronic projection and 3D printing may seem significant, they can be rectified (as shown in [6]) by setting the scale during 2D/3D projection from 19.685 pixels/mm to 16.79 pixels/mm. In doing so, the errors introduced during mold modeling (Fig. 2.4 (a)) and 3D mold printing (Fig. 2.4 (b)) are compensated.

(iii) Error in Casting - The fidelity in the universal fingerprint target post casting (Figs. 2.4 (d), (e)) is validated in the following manner. First, three universal fingerprint target castings are fabricated using three different molds; each mapped with different 2D calibration patterns (vertical, horizontal, and circular sine gratings with a frequency of 10 pixels). At a projection scale of 16.79 pixels/mm (at 500 ppi) and the reduction in point-to-point distances during electronic modeling and 3D printing, 10 pixel ridge distances on the calibration pattern should correspond to an actual ridge distance of 0.508 mm on the casted calibration target. Using an optical microscope, 5 images of each universal fingerprint target are captured at both 50X magnification and 100X magnification (Fig. 2.11) [98]. A software tool available with the optical microscope is used to mark 20 point-to-point ridge distances at 50X magnification and 10 point-to-point ridge distances at 100X magnification in all the acquired optical microscope images. The microscope software was calibrated using a micrometer resolution calibration target. Table 2.4 shows the average point-to-point ridge distances at both magnifications for all 3 casted targets. In comparison to the ground truth distance of 0.508 mm, the optical microscope reveals the empirical mean point-to-point ridge distances to be 0.499 mm, attributing to a 1.8 % reduction in point-to-point distances on the universal fingerprint target during casting. This reduction of 1.8 % in point-to-point ridge distances is not unexpected, since the conductive silicone used to fabricate the universal fingerprint targets is estimated to shrink by 2 % during vulcanization. Again, this error can be compensated by adjusting the projection scale during 2D/3D mapping.

In addition to measuring point-to-point distances, we also measure the height of the ridges on the casted targets using a high resolution profilometer [2]. The ridge height of the fingerprint

Table 2.4 Average point-to-point ridge distances observed on universal fingerprint targets, measured using the Keyence Optical Microscope at 50X and 100X magnification. The expected point-to-point ridge distance is 0.508 mm. (standard deviation is recorded in parenthesis).

Calibration Pattern	50X Magnification	100X Magnification
Vertical Gratings	0.509 mm (.031)	0.496 mm (.023)
Horizontal Gratings	0.501 mm (.026)	0.490 mm (.028)
Circular Gratings	0.513 mm (.029)	0.486 mm (.035)

targets is set to 0.33 mm during electronic projection. Due to mold shrinkage during 3D printing, we expect the ridge height of the casted targets to be 0.29 mm. The measurements obtained by the profilometer show all ridge heights to be 0.16 mm. This further reduction in ridge height is not unexpected since a thin coating of release agent is first applied to the mold prior to casting. Furthermore, the reduction in ridge height is beneficial as it brings the ridge height of the targets even closer in value to the human finger ridges at 0.06 mm. Note, the ridge height had to be set to 0.33 mm during electronic projection due to current limitations in state-of-the-art 3D printing resolution. Future study could explore novel techniques for fabricating the mold which enable even higher resolution than 3D printing.

(iv) End-to-end Error- In this final error analysis, the full, end-to-end fabrication process is scrutinized. More specifically, an experiment is conducted which demonstrates that features present on a 2D fingerprint image are preserved after converting the 2D fingerprint image into a wearable, 3D, universal fingerprint target.

To conduct this experiment, six different universal fingerprint target molds are fabricated using six fingerprint images from the NIST SD4 database [129]. Subsequently, six universal fingerprint targets are cast from the fingerprint molds. Finally, comparison scores are generated between the NIST SD4 rolled fingerprint images and 2D fingerprint images acquired from the corresponding six universal fingerprint targets. Fingerprint images of the universal fingerprint targets are obtained using an Appendix F certified, 500 ppi, contact-optical reader, a PIV certified, 500 ppi, contactless-optical reader, and a PIV certified, 500 ppi, capacitive reader. Figure 2.12 illustrates corresponding minutia points between a NIST SD4 rolled fingerprint image and a fingerprint image acquired from

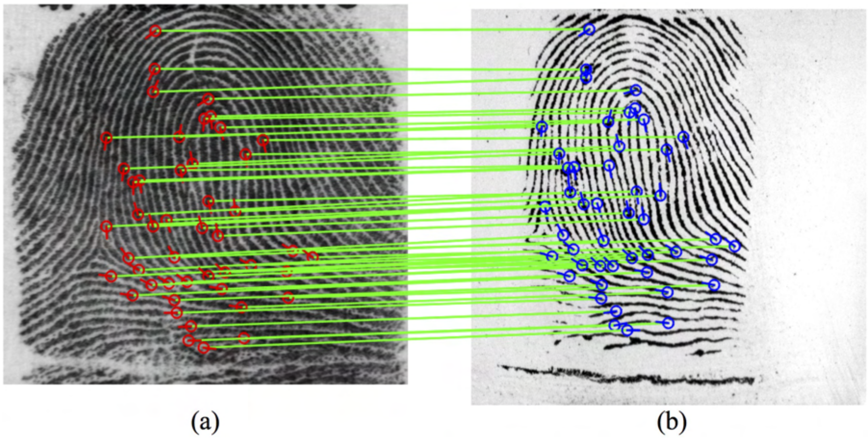


Figure 2.12 Comparing the source fingerprint image to the image of the corresponding universal fingerprint target. (a) NIST SD4 S0083 rolled fingerprint image is compared to (b) a universal fingerprint target image; (b) is fabricated using (a) and imaged using an Appendix F certified, optical, 500 ppi fingerprint reader. A similarity score of 608 is computed between (a) and (b) using Verifinger 6.3 SDK (threshold is 33 at FAR=0.01 %). The minutia points in correspondence between (a) and (b) are shown.

its corresponding universal fingerprint target. Table 2.5 reports similarity scores for each of the six universal fingerprint targets in comparison to the NIST SD4 rolled print used to fabricate them.

The key findings of this experiment are as follows:

- The corresponding minutia points between images captured using the universal fingerprint targets and the images used to generate each target (Fig. 2.12) show that salient 2D features inherent to the NIST rolled fingerprint images are retained following their fabrication into a universal fingerprint target.

- The universal fingerprint targets (Table 2.5) almost always outperform previous 3D optical targets [5] (Table 2.6) by achieving higher similarity scores between the finished 3D target images and the ground truth image used to fabricate the respective target. Furthermore, the universal fingerprint targets perform comparably to goldfingers [6] on capacitive readers (Tables 2.5 and 2.6).

- Unlike past research in 3D fingerprint targets, the universal fingerprint target achieves comparison scores on contactless-optical readers well above the acceptance threshold[8] of 33. We

[8] We use Verifinger 6.3 which has a threshold of 33 at a FAR=0.01%.

Table 2.5 Universal Fingerprint Target Similarity Scores[1] (SD4 fingerprint image vs. corresponding target image). Proposed Targets.

SD4 Fingerprint	Contact Optical (500 ppi)	Contactless Optical (500 ppi)	Capacitive (500 ppi)
S0005	584	152	161
S0010	539	137	305
S0031	600	105	221
S0044	498	150	323
S0068	327	146	368
S0083	608	176	323

[1] Verifinger 6.3 SDK was used for generating similarity scores. The score threshold at 0.01 % FAR is 33. Verifinger was chosen so that comparisons could be made between the universal fingerprint targets and previous studies [5] [4] [6]

do note that the universal fingerprint targets achieve lower comparison scores against the SD4 images when using the contactless-optical reader as opposed to the contact-optical reader for image acquisition. One plausible explanation is that the universal fingerprint targets have a ridge height greater than the ridge height of the adult human finger. This discrepancy may cause errors as the contactless-reader unrolls a 3D fingerprint into a 2D fingerprint image.

In summary, the 2D ground truth fingerprint features are found to be preserved during fabrication into a 3D universal fingerprint target and subsequent image acquisition (with high accuracy) by contact-based optical readers, contactless-optical readers, and capacitive readers, as evidenced by the high minutiae-based match scores.

2.4.2 Reproducibility

In the previous section, the fabrication process for creating universal fingerprint targets was quantitatively shown to be of high fidelity. One remaining criterion that must be objectively verified to solidify the use of universal fingerprint targets as standardized evaluation artifacts is the re-

Table 2.6 3D Printed Target[1] Similarity Scores (SD4 fingerprint image vs. corresponding target image). Targets from [4–6].

SD4 Fingerprint	Contact-Optical Reader (500 ppi) [4,5]	Capacitive Reader (500 ppi) [6]
S0005	719	471
S0010	129	333
S0031	N/A	N/A
S0044	371	N/A
S0068	N/A	N/A
S0083	441	183

[1] These targets were fabricated using processes reported in [4–6]. They are not interoperable across optical and capacitive readers as are the Universal Fingerprint Targets.

producibility of high fidelity universal fingerprint target fabrication. To that end, we individually examine the reproducibility of each step in the universal fingerprint target fabrication process.

The electronic model of the universal fingerprint target mold and scaffolding system can be easily reproduced by simply executing a program. Additionally, the mold and scaffolding system can be physically reproduced via 3D printing with accuracy as high as 20 microns [162]. Therefore, the only step in the universal fingerprint target fabrication process that must still be verified as reproducible is the casting step.

To demonstrate reproducibility in casting, 12 universal fingerprint targets are fabricated from 6 fingerprint molds. The 12 universal fingerprint targets correspond to 6 different targets each fabricated 2 times (with a time lapse of several weeks between target replication). Each mold is mapped with one of 6 NIST SD4 rolled fingerprint images (S0005, S0010, S0031, S0044, S0068, and S0083). Let the two sets of universal fingerprint targets be formally defined as T_1 and T_2, where T_1 is the first set of castings and T_2 is the set of castings produced several weeks later.

Next, the average and standard deviation of genuine scores between 10 impressions from each target in the two target sets T_1 and T_2 collected on 3 types of fingerprint readers (*COR_A*, *CLOR*, and *CPR_A* (Table 2.8)) and the corresponding fingerprint image in NIST SD4 are computed using

Table 2.7 Universal Fingerprint Target Genuine Similarity Scores[1] (SD4 Fingerprint Image vs. Corresponding Target Image) Mean and (Standard Deviation) of Scores for 10 Impressions are Reported

Target Set	Reader	S0005	S0010	S0031	S0044	S0068	S0083
T_1	*COR_A*	212.2 (20.0)	200.3 (15.8)	204.4 (19.3)	177.0 (23.1)	141.3 (9.0)	254.5 (20.1)
T_2	*COR_A*	247.4 (10.3)	207.0 (9.2)	226.8 (17.3)	230.6 (16.0)	166.5 (10.0)	248.7 (8.4)
T_1	*CLOR*	203.9 (15.8)	127.5 (15.2)	140.6 (7.8)	154.3 (11.8)	169.9 (17.3)	172.8 (17.8)
T_2	*CLOR*	205.1 (14.1)	134.3 (23.0)	150.7 (10.3)	143.4 (15.7)	170.5 (22.2)	172.0 (11.5)
T_1	*CPR_A*	163.2 (21.8)	128.6 (25.2)	177.1 (14.5)	141.3 (22.7)	121.3 (14.2)	190.9 (17.1)
T_2	*CPR_A*	188.1 (16.7)	183.8 (16.5)	194.4 (21.6)	173.3 (10.4)	156.5 (7.0)	194.0 (6.7)

[1] Innovatrics matcher was used to generate similarity scores. The threshold of the matcher at FAR = 0.01 % was computed to be 49 on the FVC 2002 and 2004 databases [108, 109].

the Innovatrics fingerprint SDK[9] [80]. The averages and standard deviations of genuine similarity scores between target impressions from each target in T_1 and its corresponding fingerprint image in NIST SD4 are formally defined as GS_1. Conversely, GS_2 is defined as the averages and standard deviations of genuine similarity scores between target impressions for each target in T_2 and its corresponding fingerprint image in NIST SD4.

By analyzing the means of the similarity scores in GS_1 and GS_2, reproducibility in casting universal fingerprint targets is verified. In particular, by showing that the means of the similarity scores in GS_1 and GS_2 are all well above the genuine acceptance threshold, we demonstrate that targets (from multiple castings) in T_1 and T_2 are all of high fidelity, since impressions from both sets of targets (on multiple types of fingerprint readers) achieve high similarity scores against the

[9] We use the Innovatrics fingerprint SDK since we recently acquired this matcher, and it is shown to have high accuracy. Mention of any products or manufacturers does not imply endorsement by the authors or their institutions of these products or their manufacturers.

ground truth images (SD4) from which they were fabricated. The means and standard deviations of the genuine similarity scores in GS_1 and GS_2 are reported in Table 2.7.

We note that the means of all similarity scores in GS_1 and GS_2 are within 0.72 $\%$ when using the contactless fingerprint reader for image acquisition (Table 2.7). This indicates high similarity between 3D fingerprint topologies on targets in T_1 and T_2. Additionally we note that the means of similarity scores in GS_1 and GS_2 differ slightly when using contact based fingerprint readers for image acquisition. This is not surprising since the targets in T_1 were fabricated with smaller amounts of silicone thinner than the targets in T_2. As such, the softer targets in T_2 morphed around the fingerprint reader platen more than the targets in T_1 and produced images with larger friction ridge area and number of minutia (recall Fig. 2.9 (b)). Subsequently, the larger fingerprint images acquired from targets in T_2 achieved higher match scores against SD4 images than fingerprint images acquired from targets in T_1. This finding underscores one of the key advantages of contactless fingerprint readers. In particular, it shows that contactless readers are robust to small mechanical variations in human finger epidermis.

Table 2.8 Specifications of the Fingerprint Readers Used in Our Experiments

Reader NDA Alias[1]	Reader Type	Resolution	Certifications
COR_A	Contact-Optical	500 *ppi*	Appendix F
COR_B	Contact-Optical	500 *ppi*	Appendix F
$CLOR$	Contactless-Optical	500 *ppi*	PIV
CPR_A	Capacitive	500 *ppi*	PIV
CPR_B	Capacitive	500 *ppi*	PIV

[1] Because of a Nondisclosure agreement (NDA) with our vendors, we do not release the names of the fingerprint readers.

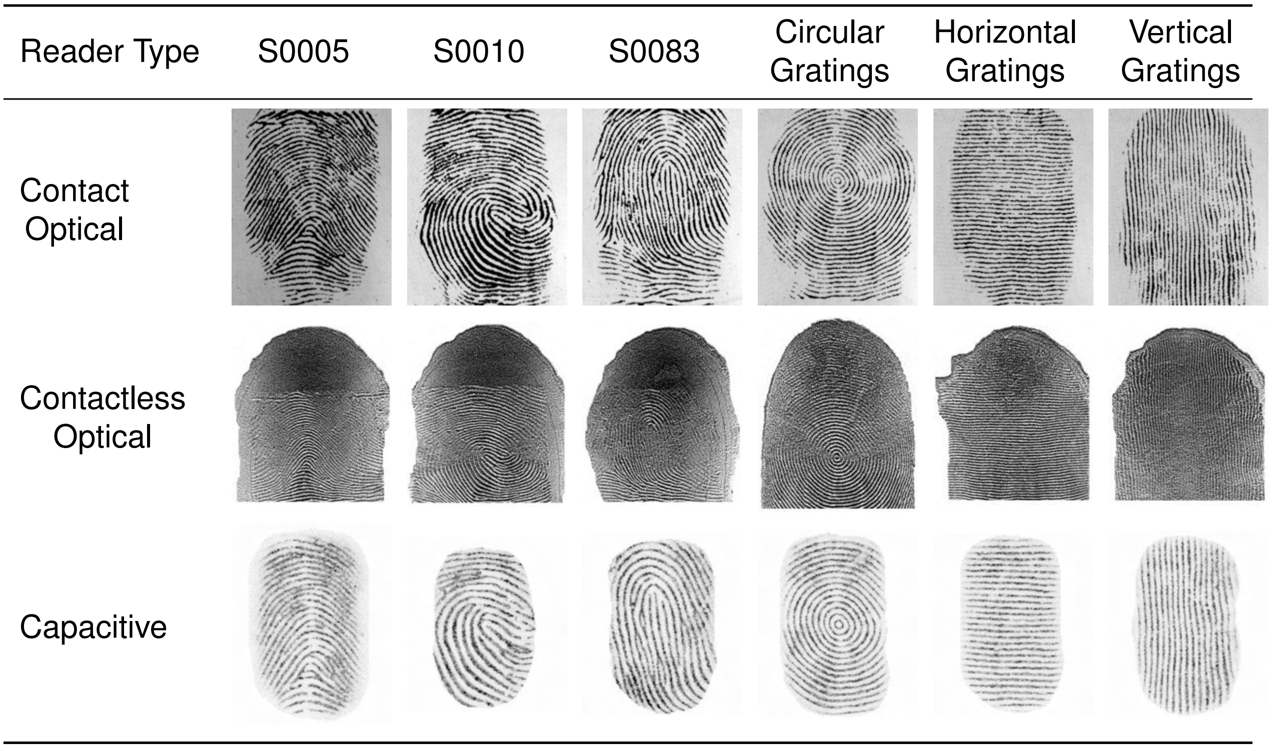


Figure 2.13 Example fingerprint impressions from 6 universal fingerprint targets (one per column) on 3 types of fingerprint readers.

2.5 Experiments

With the fidelity and reproducibility of the universal fingerprint target fabrication process established, multiple experiments are performed on all three major types of fingerprint readers using universal fingerprint targets as operational evaluation targets. First, three fingerprint readers (*COR_A, CLOR*, and *CPR_A* (Table 2.8)) are individually assessed using three different universal fingerprint targets mapped with controlled calibration patterns (horizontal gratings, vertical gratings, and circular gratings). Next, the same three fingerprint readers are individually evaluated using impressions acquired from fingerprint targets in T_2. Finally, a fingerprint reader interoperability study is performed by comparing images acquired from one of three reader types (contact-optical, contactless-optical, and capacitive) against images acquired from another of the three reader types.

Table 2.9 Mean (μ) and std. deviation (σ) of center-to-center ridge spacings (in pixels) on images acquired from 3 universal fingerprint targets. The expected ridge spacing is 9.8 pixels.

Sine Gratings Pattern	Contact Optical (*COR_A*)	Contactless Optical (*CLOR*)	Capacitive (*CPR_A*)
Circular	$\mu = 9.50$ $\sigma = 0.56$	$\mu = 8.99$ $\sigma = 0.06$	$\mu = 9.75$ $\sigma = 0.12$
Horizontal	$\mu = 9.21$ $\sigma = 0.65$	$\mu = 8.94$ $\sigma = 0.16$	$\mu = 9.45$ $\sigma = 0.10$
Vertical	$\mu = 8.90$ $\sigma = 0.88$	$\mu = 7.63$ $\sigma = 0.51$	$\mu = 9.17$ $\sigma = 0.09$

2.5.1 Evaluating Readers with Calibration Patterns

To evaluate the directional imaging capability of fingerprint readers, we design a similar experiment to that which is proposed in [5]. In particular, we collect 10 impressions on 3 different types of fingerprint readers using 3 different universal fingerprint targets mapped with controlled calibration patterns (example impressions shown in Fig. 2.13). Then, using the method in [74] the average ridge-to-ridge spacing (in pixels) is computed for the captured impressions. Unlike the targets proposed in [4–6] which could only perform directional assessment of one type of fingerprint reader, our proposed universal fingerprint targets are capable of performing directional assessment on contact-optical, contactless-optical, and capacitive fingerprint readers alike. Therefore, in Table 2.9, we report the average ridge-to-ridge spacing of the 3 different universal fingerprint targets across all three of the major fingerprint reader types. For comparison to previous work [4–6], the ridge spacing values acquired from sine grating mapped, 3D printed targets (using several fabrication processes) are reported in Table 2.10.

All three of the calibration patterns that were mapped to universal fingerprint targets have a 10 pixel peak-to-peak frequency. Given our earlier findings of an approximately 2 % decrease in point-to-point distances on the universal fingerprint targets during fabrication (due to silicone shrinkage), ridge-to-ridge distances on the 3 calibration mapped universal fingerprint targets are

Table 2.10 Mean (μ) and std. deviation (σ) of center-to-center ridge spacings (in pixels) on images acquired from 3D printed targets[1]. The expected ridge spacing is 8.28 pixels.

Sine Gratings Pattern	Contact Optical [5]	Contactless Optical [4]	Capacitive [6][2]
Circular	$\mu = 8.92$ $\sigma = 0.04$	$\mu = 8.12$ $\sigma = 0.16$	N/A
Horizontal	$\mu = 8.31$ $\sigma = 0.10$	N/A	N/A
Vertical	$\mu = 8.87$ $\sigma = 0.08$	N/A	N/A

[1] These targets were fabricated using processes reported in [4–6]. They are not interoperable across optical and capacitive readers as are the Universal Fingerprint Targets.

[2] No ridge spacing results were reported for sine grating mapped gold fingers in [6].

expected to be 9.8 pixels. Given this ground truth value and the results of Table 2.9, we can evaluate the three types of fingerprint readers used in this experiment.

The summary of our findings are as follows:

- Similar to the findings of [5], impressions of targets mapped with circular gratings have larger ridge-to-ridge spacing than impressions of targets mapped with horizontal or vertical gratings. As noted in [5], this is likely due to the radial flattening of the target with circular gratings as it is applied with pressure to the fingerprint reader platen. This radial flattening results in larger ridge-to-ridge spacing than the flattening of the horizontal and vertical calibration targets.

- Unlike the findings of [5], all of the captured impressions of universal fingerprint targets have smaller ridge-to-ridge spacing than the expected ridge-to-ridge spacing. In [5] a larger than expected ridge-to-ridge spacing was explained as a result of ridge-to-ridge distance expansion during the flattening of the target against the reader platen. We hypothesize that universal fingerprint targets have smaller ridge-to-ridge expansion during contact with the reader platen than [5] since universal fingerprint targets are less elastic than the targets in [5].

Universal fingerprint targets are closer in elasticity to the human skin than [5] and so the results shown in Table 2.9 are more indicative of the ridge-to-ridge spacing the readers used in this study are able to capture from real human fingers.

- Consistent with the findings of [4], the ridge-to-ridge distances are smaller on the contactless fingerprint reader than on the contact fingerprint readers. In particular, the captured ridge-to-ridge spacing of the vertical gratings was lower than expected. We hypothesize that the ridge-to-ridge spacing on the contactless reader is smaller due to the fact that no distortion occurs during image acquisition (as no pressure is applied onto a reader platen). Further analysis needs to be undertaken to understand why the vertical gratings deviated most from the expected ridge spacing.

Table 2.11 Mean (μ) and std. deviation (σ) of center-to-center ridge spacings (in pixels) on images acquired from 6 universal fingerprint targets. Expected ridge spacing (in pixels) for each target is reported in parenthesis

SD4 Fingerprint	Contact Optical (*COR_A*)	Contactless Optical (*CLOR*)	Capacitive (*CPR_A*)
S0005 (9.25)	$\mu = 8.77$ $\sigma = 1.17$	$\mu = 8.77$ $\sigma = 0.31$	$\mu = 9.01$ $\sigma = 0.18$
S0010 (9.98)	$\mu = 9.87$ $\sigma = 1.46$	$\mu = 9.52$ $\sigma = 0.29$	$\mu = 10.42$ $\sigma = 0.41$
S0031 (10.37)	$\mu = 10.02$ $\sigma = 1.40$	$\mu = 9.04$ $\sigma = 0.37$	$\mu = 10.45$ $\sigma = 0.28$
S0044 (9.07)	$\mu = 8.49$ $\sigma = 1.24$	$\mu = 8.25$ $\sigma = 0.18$	$\mu = 9.04$ $\sigma = 0.23$
S0068 (9.48)	$\mu = 9.60$ $\sigma = 1.29$	$\mu = 9.18$ $\sigma = 0.29$	$\mu = 9.86$ $\sigma = 0.19$
S0083 (10.23)	$\mu = 9.70$ $\sigma = 1.23$	$\mu = 8.16$ $\sigma = 0.15$	$\mu = 10.23$ $\sigma = 0.14$

2.5.2 Evaluating Readers with Fingerprint Patterns

Similar to the previous experiment, we conduct an analysis of the ridge-to-ridge distances captured by three of the major fingerprint reader types. However, in this experiment, rather than mapping controlled calibration patterns to universal fingerprint targets, we use the targets from T_2 which are each mapped with real fingerprint images from SD4. In doing so, we evaluate the readers with targets very similar to the real fingers the readers will see in an operational setting.

Again, 10 impressions are captured on all 3 fingerprint readers, this time with each of the 6 universal fingerprint targets in T_2 (example impressions shown in Fig. 2.13). Then, using the method in [74], the average ridge spacing of the captured impressions is computed (Table 2.11). Additionally, the average ridge spacing is computed (using the method in [74]) on the original fingerprint images from SD4 and established as the ground truth ridge spacing values. By comparing these ground truth values with the results of Table 2.11, we perform an assessment of the three fingerprint readers. Finally, for comparison to previous work [4–6], the ridge spacing values acquired from fingerprint mapped, 3D printed targets (using several fabrication processes) are reported in Table 2.12.

In summary, the findings of this experiment are as follows:

- Consistent with the findings of our previous experiment with calibration pattern mapped universal fingerprint targets, the images captured by the contactless-optical fingerprint reader have smaller ridge-to-ridge distances than the impressions captured by contact based readers. This is likely due to the absence of fingerprint distortions in contactless fingerprint readers. Additionally, errors in the contactless reader may be introduced when the three-dimensional finger surface captured by the reader is projected into two dimensions (due to the ridge height of universal fingerprint targets being greater than the ridge height of human fingers).
- In almost all of the target impressions, capacitive fingerprint readers captured the ridge-to-ridge distances more closely to ground truth than contact-optical readers did. Further studies

Table 2.12 Mean (μ) and std. deviation (σ) of center-to-center ridge spacings (in pixels) on images acquired from 3D printed fingerprint targets[1]. Expected ridge spacing (in pixels) for each target is reported in parenthesis

SD4 Fingerprint	Contact Optical [5]	Contactless Optical [4][2]	Capacitive [6]
S0005	$\mu = 8.49$ $\sigma = 0.10$ (7.82)	N/A	$\mu = 9.57$ $\sigma = 0.14$ (9.45)
S0010	$\mu = 9.22$ $\sigma = 0.16$ (8.43)	N/A	$\mu = 10.34$ $\sigma = 0.21$ (10.20)
S0083	$\mu = 9.10$ $\sigma = 0.19$ (8.62)	N/A	$\mu = 10.60$ $\sigma = 0.14$ (10.44)

[1] These targets were fabricated using different processes reported in [4–6]. They are not interoperable across optical and capacitive readers as are the Universal Fingerprint Targets.
[2] No ridge spacing results were reported for fingerprint mapped targets imaged by a contactless reader in [4].

and analysis need to be performed to determine if this finding is consistent, and also, the explanation behind this.

2.5.3 Reader Interoperability Evaluations

Whereas our previous two experiments with universal fingerprint targets evaluated the three major types of fingerprint readers individually, in this final experiment, we perform fingerprint reader *interoperability* evaluations using the universal fingerprint targets.

To set up this experiment, 10 impressions from each target in T_2 are captured on 5 different fingerprint readers (Table 2.8). Then, for all pairs of fingerprint readers in our set of 5 readers, images from one reader are used as enrollment images and images from the other reader are used as probe images to generate genuine and imposter scores using the Innovatrics matcher [80]. In Table 2.13, we report the means of the genuine and imposter scores. Additionally we report the True Accept Rate (TAR) and the False Accept Rate (FAR) of the scores using a threshold of 49

Table 2.13 Genuine and Imposter Score[1] Statistics and Matching Performance Measures when Comparing Fingerprint Images Acquired from Different Types of Fingerprint Readers. Mean of Genuine Scores (μG), Mean of Imposter Scores (μI), True Accept Rate (TAR) and False Accept Rate (FAR[2]) are Reported.

	Probe Image Fingerprint Readers				
Enrollment Reader	*COR_A*	*COR_B*	*CLOR*	*CPR_A*	*CPR_B*
COR_A	$\mu G = 440.7$, $\mu I = 0.5$ $TAR = 100\%$	$\mu G = 399.6$, $\mu I = 0.3$ $TAR : 100\%$	$\mu G = 182.2$, $\mu I = 1.2$ $TAR : 100\%$	$\mu G = 276.0$, $\mu I = 1.9$ $TAR : 100\%$	$\mu G = 202.4$, $\mu I = 4.8$ $TAR : 100\%$
COR_B	$\mu G = 399.3$, $\mu I = 0.3$ $TAR : 100\%$	$\mu G = 438.1$, $\mu I = 0.2$ $TAR : 100\%$	$\mu G = 171.3$, $\mu I = 0.5$ $TAR : 99.8\%$	$\mu G = 278.5$, $\mu I = 1.6$ $TAR : 100\%$	$\mu G = 200.0$, $\mu I = 4.3$ $TAR : 100\%$
CLOR	$\mu G = 183.8$, $\mu I = 1.4$ $TAR : 100\%$	$\mu G = 174.3$, $\mu I = 0.5$ $TAR : 100\%$	$\mu G = 334.1$, $\mu I = 9.0$ $TAR : 100\%$	$\mu G = 154.1$, $\mu I = 2.1$ $TAR : 99.8\%$	$\mu G = 113.2$, $\mu I = 4.6$ $TAR : 94.8\%$
CPR_A	$\mu G = 271.1$, $\mu I = 0.8$ $TAR : 100\%$	$\mu G = 274.7$, $\mu I = 0.8$ $TAR : 100\%$	$\mu G = 147.0$, $\mu I = 1.6$ $TAR : 99.7\%$	$\mu G = 353.0$, $\mu I = 7.6$ $TAR : 100\%$	$\mu G = 269.4$, $\mu I = 12.1$ $TAR : 100\%$
CPR_B	$\mu G = 196.4$, $\mu I = 2.3$ $TAR : 100\%$	$\mu G = 195.4$, $\mu I = 2.2$ $TAR : 100\%$	$\mu G = 105.4$, $\mu I = 3.2$ $TAR : 91.8\%$	$\mu G = 268.2$, $\mu I = 10.1$ $TAR : 100\%$	$\mu G = 277.5$, $\mu I = 14.4$ $TAR : 100\%$

[1] Innovatrics matcher was used to generate similarity scores. The threshold of the matcher at FAR = 0.01 % was computed to be 49 on the FVC 2002 and 2004 databases [108, 109].

[2] The False Accept Rate in all cases was 0.0%

(this threshold was precomputed on the FVC 2002 and 2004 databases [108, 109], because we do not have a sufficient number of images from the targets to set the threshold).

Although the performance results of Table 2.13 seem to indicate that all of the readers used are highly interoperable, these results are likely too optimistic as only 6 different targets were used. For this reason, we also report the genuine and imposter score means to show how the scores deteriorate when different readers are used for enrollment and verification. Similar to the findings of past fingerprint reader interoperability studies [120, 144, 145], we note that genuine scores decrease and imposter scores increase when different fingerprint readers are used to acquire enrollment images and probe images, especially when the two readers use different sensing technology to acquire images. While past studies reported these findings using real fingers for data collection, we report the same findings, for the first time ever, using realistic, 3D, wearable, fingerprint targets. By demonstrating the same results as past studies with the universal fingerprint targets, we validate the utility in using universal fingerprint targets for advancing fingerprint reader interoperability studies. In particular, the universal fingerprint targets could be mounted to a robot and imaged on different readers at known pressure and orientation. This standardized data could then be used to learn calibration mappings between different fingerprint readers which could be used to improve fingerprint reader interoperability.

2.6 Summary

We have designed a molding and casting system capable of fabricating wearable, 3D fingerprint targets from a plethora of casting materials. By selecting a casting material with similar mechanical, optical, and electrical properties to the human skin, we cast *universal fingerprint targets*, which can be imaged on the three major fingerprint reader types in use (contact-optical, contactless-optical, and capacitive). Previous studies were unable to produce a single 3D fingerprint target which could be imaged on multiple types of fingerprint readers. We demonstrate that the process for fabricating universal fingerprint targets is of high fidelity, and that it is reproducible. Finally, we use the universal fingerprint targets as evaluation targets on multiple types of PIV/Appendix F certified fingerprint readers. Our results verify the utility in using the universal fingerprint targets for both individual fingerprint reader assessments and also fingerprint reader interoperability studies. We believe that the universal 3D fingerprint targets introduced here will advance state of the art in fingerprint reader evaluation and interoperability studies.

In the future, the universal fingerprint targets will be mounted to a robotic hand and imaged on various fingerprint readers at known pressure and orientation. With this data, objective evaluations can be performed on fingerprint readers. Additionally, the data collected could be utilized to learn fingerprint distortion models, fingerprint reader interoperability calibration models, and latent fingerprint distortion models. Finally, the universal fingerprint targets will be used to assess the spoofing vulnerability of various fingerprint recognition systems (such as smartphones).

2.7 Acknowledgment

This research was supported by grant no. 70NANB15H280 from the NIST Measurement Science program. 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 the U.S. Government.

The authors would like to thank Brian Wright, Michigan State University, for his help in 3D printing of molds. We would also like to thank Edward Drown, Michigan State University, for his help mixing castings materials for the universal fingerprint targets. Finally, we would like to thank Nick Paulter for his valuable insights throughout the research.

Chapter 3

RaspiReader: Open Source Fingerprint Reader

In the previous chapter, we demonstrated a manufacturing process for creating high-fidelity, realistic, universal, 3D fingerprint targets. While we proposed the use of these targets for evaluating fingerprint readers, it has been shown that hackers can use fingerprint targets, like the Universal Target, to "spoof" fingerprint recognition systems. That is, hackers can use fake fingerprints (also known as presentation attacks, or spoofs) to impersonate a victim or to obfuscate their own identity. To thwart such attacks, in this chapter, we develop a fingerprint reader, called RaspiReader, with the built-in capability of automatically detecting and flagging fingerprint spoof attacks prior to performing recognition [47]. RaspiReader is an open-source, easy to assemble, high resolution, optical fingerprint reader, built entirely from ubiquitous components. More importantly, RaspiReader is specially customized with two cameras for fingerprint image acquisition. One camera provides high contrast, frustrated total internal reflection (FTIR) fingerprint images, and the other outputs direct images of the finger in contact with the platen. Using both of these image streams, we extract complementary information which, when fused together and used for spoof detection, results in marked performance improvement over previous methods relying only on grayscale FTIR images provided by COTS optical readers. Fingerprint matching experiments between images acquired

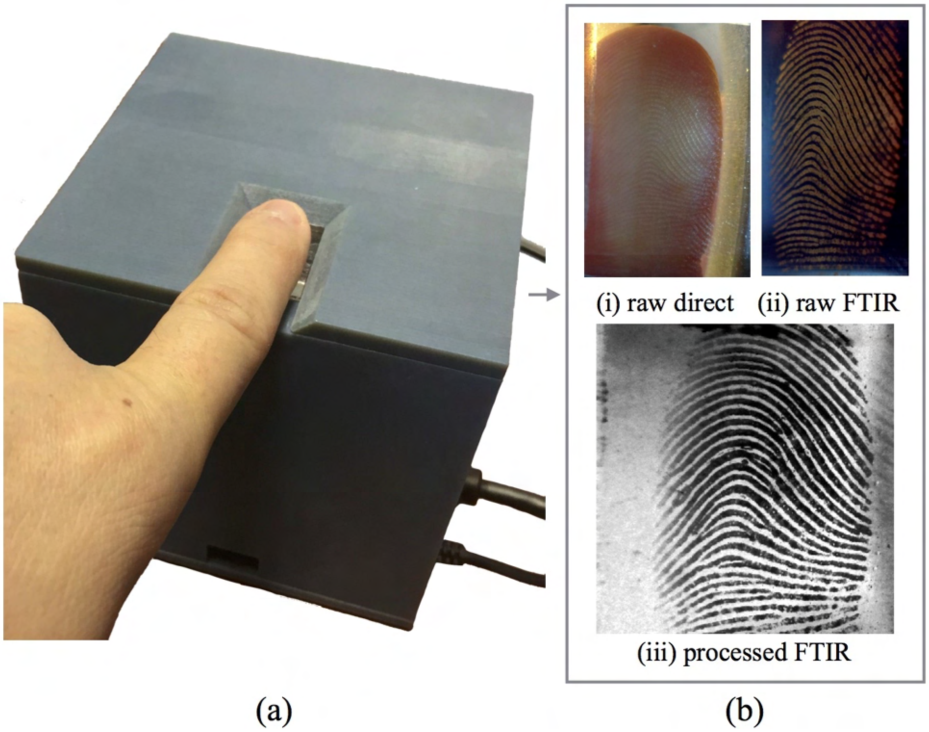


Figure 3.1 Prototype of RaspiReader: two fingerprint images (b, (i)) and (b, (ii)) of the input finger (a) are captured. The raw direct image (b, (i)) and the raw, high contrast FTIR image (b, (ii)) both contain useful information for spoof detection. Following the use of (b, (ii)) for spoof detection, image calibration and processing are performed on the raw FTIR image to output a high quality, 500 ppi fingerprint for matching (b, (iii)). The dimensions of the RaspiReader shown in (a) are 100 *mm* x 100 *mm* x 105 *mm* (about the size of a 4 inch cube).

from the FTIR output of RaspiReader and images acquired from a COTS reader verify the interoperability of the RaspiReader with existing COTS optical readers. By using our open source STL files and software, RaspiReader can be built in under one hour for only US $175.

3.1 Introduction

In an effort to mitigate the costs associated with fingerprint spoof attacks, a number of spoof detection techniques involving both hardware and software have been proposed in the literature. Special hardware embedded in fingerprint readers[1] enables capture of features such as heartbeat,

[1] Several fingerprint vendors have developed hardware spoof detection solutions by employing multispectral imaging, infrared imaging (useful for sub-dermal finger analysis), and pulse capture to distinguish live fingers from spoof fingers [67, 130].

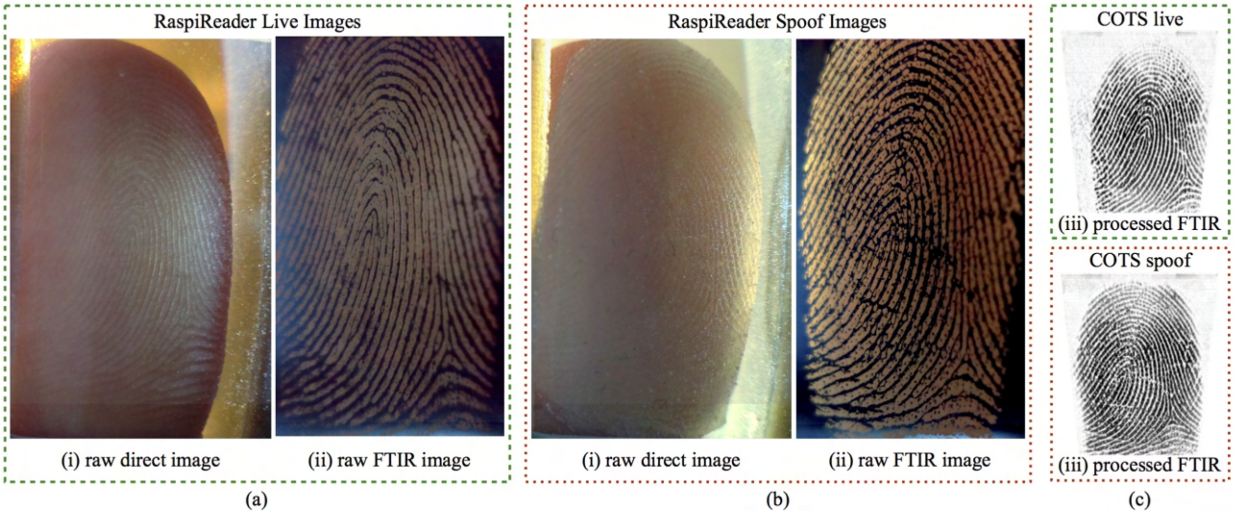


Figure 3.2 Fingerprint images acquired using the RaspiReader. Images in (a) were collected from a live finger. Images in (b) were collected from a spoof finger. Using features extracted from both raw image outputs ((i), direct) and ((ii), FTIR) of the RaspiReader, our spoof detectors are better able to discriminate between live fingers and spoof fingers. The raw FTIR image output of the RaspiReader (ii) can be post processed (after spoof detection) to output images suitable for fingerprint matching. Images in (c) were acquired from the same live finger (a) and spoof finger (b) on a commercial off-the-shelf (COTS) 500 ppi optical reader. The close similarity between the two images in (c) qualitatively illustrates why current spoof detectors are limited by the low information content, processed fingerprint images (c, (iii)) output by COTS readers.

thermal output, blood flow, odor, and sub-dermal finger characteristics useful for distinguishing a live finger from a spoof [8, 100, 113, 130, 146, 151–153, 176]. Spoof detection methods in software are based on extracting textural [65, 66, 69, 70, 127], anatomical [64], and physiological [1, 114] features from processed[2] fingerprint images which are used in conjunction with a classifier such as Support Vector Machines (SVM). Alternatively, a Convolutional Neural Network (CNN) can be trained to distinguish a live finger from a spoof [26, 27, 117, 131].

While existing hardware and software spoof detection schemes provide a reasonable starting point for solving the spoof detection problem, current solutions have a plethora of shortcomings. As noted in [151, 152, 176] most hardware based approaches can be easily bypassed by developing very thin spoofs (Fig. 3.3 (a)), since heartbeat, thermal output, and blood flow can still be read from

[2] Raw fingerprint images are "processed" (such as RGB to grayscale conversion, contrast enhancement, and scaling) by COTS readers to boost matching performance. However, useful spoof detection information (such as color and/or minute textural abberations) is lost during this processing.

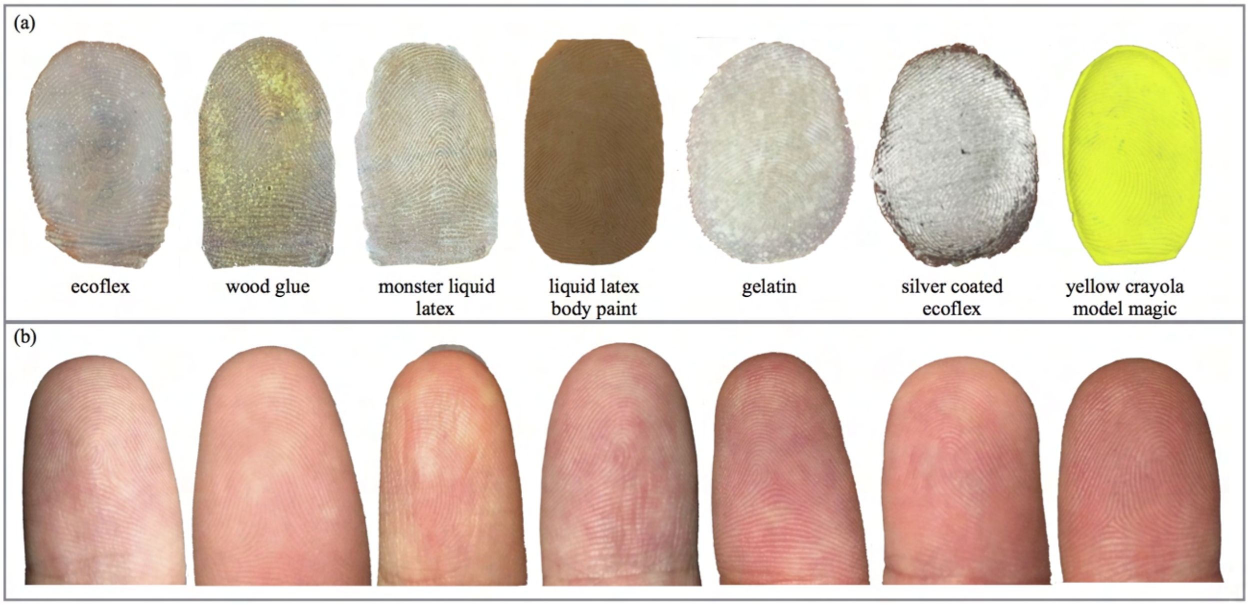


Figure 3.3 Example spoof fingers and live fingers in our database. (a) Spoof fingers and (b) live fingers used to acquire both spoof fingerprint impressions and live fingerprint impressions for conducting the experiments reported in this thesis. The spoofs in (a) and the live fingers in (b) are not in 1-to-1 correspondence.

the live human skin behind the thin spoof. Additionally, some of the characteristics (such as odor and heartbeat) acquired by the hardware vary tremendously amongst different human subjects, making it very difficult to build an adequate model representative of all live subjects [152, 176].

Current spoof detection software solutions have their own limitations. Although the LivDet 2015 competition reported state-of-the-art spoof detection software to have an average accuracy of 95.51% [121], the spoof detection performance at desired operating points such as False Detect Rate (FDR) of 0.1% was not reported, and very limited evaluation was performed to determine the effects of testing spoof detectors with spoofs fabricated from materials not seen during training (cross-material evaluation). In the limited cross material evaluation that was performed, the rate of spoofs correctly classified as spoofs was shown to drop from 96.57% to 94.20% [121]. While this slight drop in accuracy seems promising, without knowing the performance at field conditions, namely False Detect Rate (FDR)[3] of 0.1% on a larger collection of unknown materials, the reported levels of total accuracy should be accepted with caution. Chugh et al. [26] pushed

[3] The required operating point for the ODIN program supporting this research is FDR = 0.2%

state-of-the-art fingerprint spoof detection performance on the LivDet 2015 dataset from 95.51% average accuracy to 98.61% average accuracy using a CNN trained on patches around minutiae points, but they also demonstrated that performance at strict operating points dropped significantly in some experiments. For example, Chugh et al. reported an average accuracy on the LivDet 2011 dataset of 97.41%, however, at a FDR of 1.0%, the TDR was only 90.32%, indicating that current state-of-the-art spoof detection systems leave room for improvement at desired operating points. Finally, several other studies have reported up to a three-fold increase in error when testing spoof detectors on unknown material types [112, 167, 185].

Because of the less than desired performance of spoof detection software to adapt to spoofs fabricated from unseen materials, studies in [143], [142], and [38] developed open-set recognition classifiers to better detect spoofs fabricated with novel material types. However, while these classifiers are able to generalize to spoofs made with new materials better than closed-set recognition algorithms, their overall accuracy (approx. 85% - 90%) still does not meet the desired performance for field deployments. Other attempts to bridge the gap between seen and unseen material spoof detection performance include synthetic spoof generation, and adversarial representation learning [28, 71].

Given the limitations of state-of-the-art fingerprint spoof detection (both in hardware and software), it is evident that much work remains to be done in developing robust and generalizable spoof detection solutions. We posit that one of the biggest limitations facing the most successful spoof detection solutions to date (such as use of textural features [185] and CNNs [26, 117, 131]), is the processed COTS fingerprint reader images used to train spoof detectors. In particular, because COTS fingerprint readers output fingerprint images which have undergone a number of image processing operations (in an effort to achieve high matching performance), they are not optimal for fingerprint spoof detection, since valuable information such as color and textural aberrations is lost during the image processing operations. By removing color and minute textural details from the raw fingerprint images, spoof fingerprint impressions and live fingerprint impressions (acquired on

COTS optical readers) appear very similar (Fig. 3.2 (c)), even when the physical live/spoof fingers used to collect the respective fingerprint impressions appear very different (Fig. 3.3).

This limitation inherent to many existing spoof detection solutions motivated us to develop a custom, optical fingerprint reader, called RaspiReader, with the capability to output 2 raw images (from 2 different cameras) for spoof detection. By mounting two cameras at appropriate angles to a glass prism (Fig. 4.1), one camera is able to capture high contrast FTIR fingerprint images (useful for both fingerprint spoof detection and fingerprint matching) (Fig. 3.2 (ii)), while the other camera captures direct images of the finger skin in contact with the platen (useful for fingerprint spoof detection) (Fig. 3.2 (i)). Both images of the RaspiReader visually differentiate between live fingers and spoof fingers much more than the processed fingerprint images output by COTS fingerprint readers (Fig. 3.2 (c)).

RaspiReader's two camera approach is similar to that which was prescribed by Rowe et al. in [130, 146] where both an FTIR image and a direct view image were acquired using different wavelength LEDs, however, the commercial products developed around the ideas in [130, 146] act as a proprietary black box outputting only a single processed composite image of a collection of raw image frames captured under various wavelengths. As such, fingerprint researchers cannot implement new spoof detection schemes on the individual raw frames captured by the reader. Furthermore, unlike the patented ideas in [130], RaspiReader is built with ubiquitous components and open source software packages, enabling fingerprint researchers to very easily prototype their own RaspiReader, further customize it with new spoof detection hardware, and gain direct access to the raw images captured by the reader. In short, the low cost ($175 USD) and easy to implement (1 hour build time) RaspiReader is a truly unique concept which we posit will push the boundaries of state-of-the-art fingerprint spoof detection, by facilitating spoof detection schemes which use both hardware and software.

Experiments demonstrate that by utilizing the two cameras of RaspiReader, we are able to significantly boost the performance of state-of-the-art spoof detectors previously trained on COTS grayscale images (both on known-material and cross-material testing scenarios). In particular,

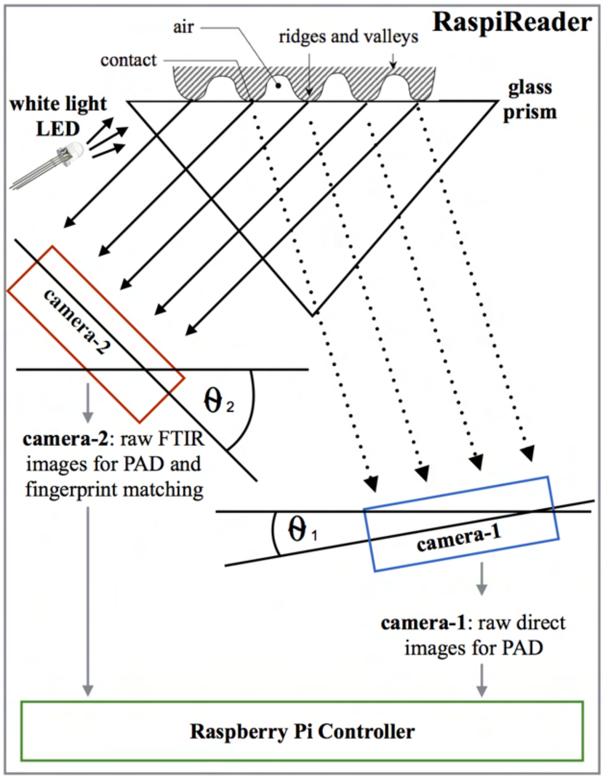


Figure 3.4 Schematic illustrating RaspiReader functionality. Incoming white light from three LEDs enters the prism. Camera 2 receives light rays reflected from the fingerprint ridges only (light rays are not reflected back from the fingerprint valleys due to total internal reflection (TIR)). This image from Camera 2, with high contrast between ridges and valleys can be used for both spoof detection and fingerprint matching. Camera 1 receives light rays reflected from both the ridges and valleys. This image from Camera 1 provides complementary information for spoof detection.

because both image outputs of the RaspiReader are raw and contain useful color information, we can extract discriminative and complementary information from each of the image outputs. By fusing this complementary information (at a feature level or score level) the performance of spoof detectors is significantly higher than when features are extracted from COTS grayscale images.

Finally, by calibrating and processing the FTIR image output of the RaspiReader (post spoof detection), we demonstrate that RaspiReader is not only interoperable with existing COTS optical readers but is also capable of achieving state-of-the-art fingerprint matching accuracy. Note that interoperability with existing COTS readers is absolutely vital in any new hardware based spoof detection solution as it makes the spoof resistant device compatible (in terms of matching) with legacy fingerprint databases[4]. Furthermore, by making the RaspiReader compatible with existing COTS readers, we further extend the utility of RaspiReader beyond spoof detection. In particular, RaspiReader is not only useful for providing direct access to multiple raw images for spoof detection; it also provides researchers in fingerprint matching the easy ability to fine tune (resolution and processing) the images being output by the fingerprint reader. In any imaging system, the recognition performance depends on the quality of the image output by the sensor. This is particularly true of fingerprint recognition systems. As shown in the NIST FpVTE 2012 [178] results, the single most important factor responsible for degrading fingerprint recognition performance is the fingerprint image quality. However, most fingerprint researchers have no control over the quality of the fingerprint images being used to develop fingerprint recognition algorithms since they must rely on blackbox COTS fingerprint readers. RaspiReader changes this by providing fingerprint matching algorithm designers an easy method for prototyping their own fingerprint reader and optimizing fingerprint image quality and fingerprint matching algorithms jointly in an effort to further improve fingerprint recognition performance.

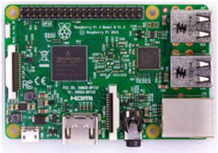
In summary, our work on RaspiReader removes the mystery of designing and understanding the internals of a fingerprint reader. Using the open-source fabrication process of this fingerprint reader, any fingerprint algorithm designer can quickly and affordably construct his or her own reader with the capabilities (spoof detection and matching image quality) necessary to meet their application requirements.

More concisely, the contributions of this chapter are:

- An open source, easy to assemble, cost effective fingerprint reader, called RaspiReader, capable of producing fingerprint images useful for spoof detection and that are of high quality and resolution (1,500 ppi - 3,300 ppi native resolution) for fingerprint matching. The custom RaspiReader can be easily modified to facilitate spoof detection and fingerprint matching studies.

[4] Interoperability with existing COTS readers is a strict requirement of the IARPA ODIN program supporting this research [76].

Table 3.1 Primary Components Used to Construct RaspiReader. Total Cost is $175.20

Component Image	Name and Description	Quantity	Cost (USD)[1]
	**Raspberry Pi 3B:** A single board computer (SBC) with 1.2 GHz 64-bit quad-core CPU, 1 GB RAM, MicroSDHC storage, and Broadcom VideoCore IV Graphic card	1	$38.27
	Raspberry Pi Camera Module V1: A 5.0 megapixel, 30 frames per second, fixed focal length camera	2	$13.49
	Multi-Camera Adapter: Splits Raspberry Pi camera slot into two slots, enabling connection of two cameras	1	$49.99
	LEDs: white light, 5 mm, 1 watt	3	$0.10
	Resistors: 1 kΩ	3	$5.16
	Right Angle Prism:[2] 25 mm leg, 35.4 mm hypotenuse	1	$54.50

[1] All items except the glass prism were purchased for the listed prices on Amazon.com

[2] The glass prism was purchased from ThorLabs [169].

- A customized fingerprint reader with two cameras for image acquisition rather than a single camera. Use of two cameras enables robust fingerprint spoof detection, since we can extract features from two complementary, information rich images instead of processed grayscale images output by traditional COTS optical fingerprint readers.

- A significant boost in spoof detection performance (both known-material and seven cross-material testing scenarios) using current state-of-the-art software based spoof detection methods in conjunction with RaspiReader images as opposed to COTS optical grayscale images. Spoofs of seven materials were used in both known-material and cross-material testing scenarios.

- Demonstrated matching interoperability of RaspiReader with a COTS optical fingerprint reader. Since RaspiReader is shown to be interoperable with COTS readers, it could immediately be deployed in the real world since interoperability makes the device compatible with legacy fingerprint databases.

3.2 RaspiReader Construction and Calibration

In this section, the construction of the RaspiReader using ubiquitous, off-the-shelf components (Table 1) is explained. In particular, the main steps involved in constructing RaspiReader consist of (i) properly mounting cameras (angle and position) with respect to a glass prism, (ii) fabricating a plastic case to house the hardware components, (iii) assembling the cameras and hardware within the plastic case, and (iv) writing software to capture fingerprint images with the assembled hardware. Each of these steps is described in more detail in the following subsections. Finally, we provide the steps for calibrating and processing the raw FTIR fingerprint images of the RaspiReader for fingerprint matching.

3.2.1 Camera Placement

The most important step in constructing RaspiReader is the placement (angle and position) of the two cameras capturing fingerprint images. In particular, to collect an FTIR image of a fingerprint, a camera needs to be mounted at an angle greater than the critical angle, and to collect a direct view image, a camera needs to be mounted an an angle less than the critical angle (both with respect to the platen). Here, the critical angle is defined as the angle at which total internal reflection occurs when light passes from a medium with an index of refraction n_1 to another medium with index of refraction n_2 (Eq. 3.2.1):

$$\theta_c = arcsin(\frac{n_2}{n_1}) \tag{3.2.1}$$

In the case of fingerprint sensing, the first medium is glass which has an index of refraction $n_1 = 1.5$, and the second medium is air which has an index of refraction of $n_2 = 1.0$ leading to a critical angle (θ_c) of $41.8°$. Therefore, as shown in (Fig. 4.1), we mount the direct view camera ($camera_1$) at an angle of $\theta_1 = 10°$ and we mount the FTIR camera ($camera_2$) at an angle of $\theta_2 = 45°$.

With respect to the position of each camera lens to the glass prism, there is a tradeoff between resolution and fingerprint area to consider. As the camera is moved closer to the prism, the fingerprint image resolution (pixels per inch) is increased. However, if the cameras are too close to the platen, only part of the fingerprint image is within the field of view (FOV). In constructing RaspiReader, we wanted to maximize the fingerprint image resolution, while still capturing the entire fingerprint image within the FOV. We experimentally determined that at a distance of 23 *mm* from the prism, the cameras would capture the entire fingerprint area. At closer distances, part of the fingerprint image would start to be outside the FOV. As a final step in camera placement, the focal length of the Raspicams (cameras used in RaspiReader) must be increased so that the camera will focus on the nearby glass prism (the default focus-length of the Raspicams is 1 meter; much greater than the 23 *mm* distant prism). By default, the Raspicams have a fixed-focal length of 3.6 *mm*. However, by rotating the Raspicam lens $652.5°$ counterclockwise (for the FTIR imaging camera) and $405°$ counterclockwise (for the direct imaging camera), the focal length can be slightly increased to bring the nearby fingerprint images into focus.

3.2.2 Case Fabrication

After determining the angle and position of both cameras, an outer casing (Fig. 3.5) accommodating these positions is electronically modeled using Meshlab [29] and subsequently 3D printed on a high resolution 3D printer (Stratasys Objet350 Connex)[5]. To make the fabrication process easily reproducible, the camera mounts and light source mounts are modeled in place on the front part of the fingerprint reader case (Fig. 3.5). As such, one only needs to 3D print the open-source STL

[5] We are currently investigating alternative case manufacturing methods such as CNC milling.

files and clip the LEDs and Raspicams to their respective mounts (Fig. 3.5) in order to quickly build their own RaspiReader replica.

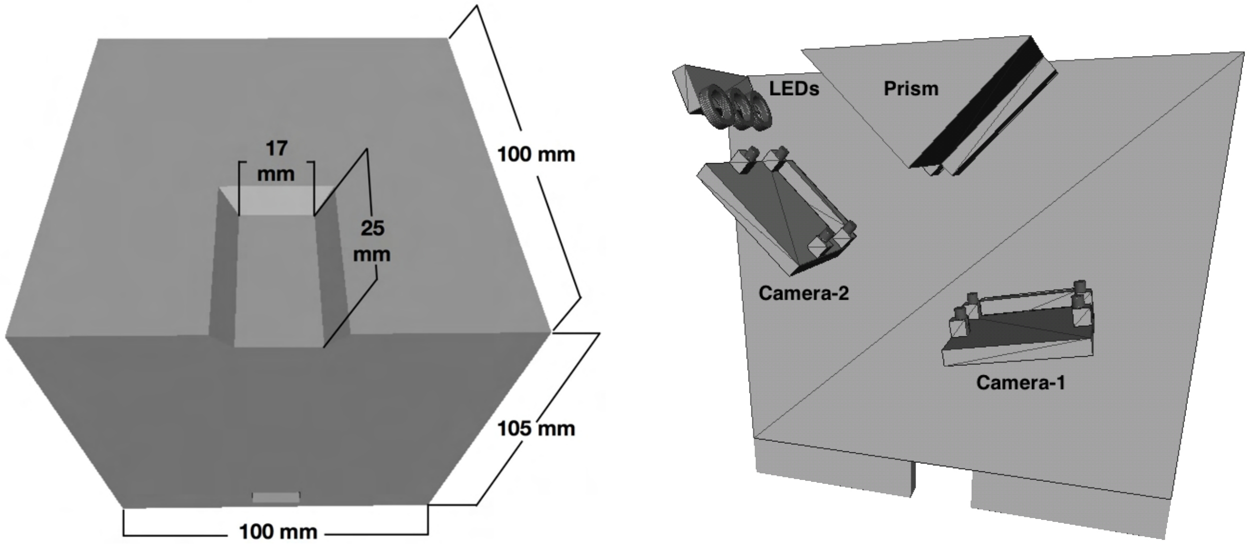


Figure 3.5 Electronic CAD model of the RaspiReader case. The camera and LED mounts are positioned at the necessary angles and distance to the glass prism, making the reproduction of RaspiReader as simple as 3D printing the open-sourced STL files.

3.2.3 Image Acquisition Hardware and Software

The backbone of the RaspiReader is the popular Raspberry Pi 3B single board computer, which enables easy interfacing with GPIO pins (for controlling LEDs) and image acquisition (with its standard camera and camera connection port). Because the Raspberry Pi only has a single camera connection port, a camera port multiplexer is used to enable the use of multiple cameras on a single Pi [3]. Using the Raspberry Pi GPIO pins, the code available in [3], and the camera multiplexer, one can easily extend the Raspberry Pi to use multiple cameras.

After assembling the camera port multiplexer to the Pi (with two Raspicams), wiring 3 LEDs to the Raspberry Pi GPIO pins, and attaching the Raspicams and LEDs to the 3D printed casing mounts (Fig. 3.5), open source python libraries [3] can be used to illuminate the glass prism and subsequently acquire two images (Fig. 3.2 (a)) from the fingerprint reader (one raw FTIR fingerprint image and another raw direct fingerprint image).

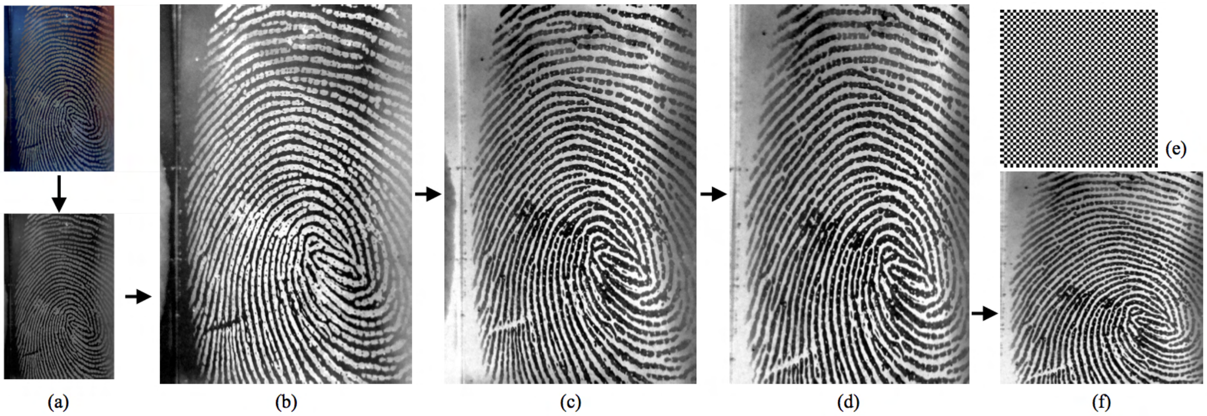


Figure 3.6 Processing a RaspiReader raw FTIR fingerprint image into a 500 ppi fingerprint image compatible for matching with existing COTS fingerprint readers. (a) The RGB FTIR image is first converted to grayscale. (b) Histogram equalization is performed to enhance the contrast between the fingerprint ridges and valleys. (c) The fingerprint is negated so that the ridges appear dark, and the valleys appear white. (d), (f) Calibration (estimated using the checkerboard calibration pattern in (e)) is applied to frontalize the fingerprint image to the image plane and down sample (by averaging neighborhood pixels) to 500 ppi in both the x and y axis.

3.2.4 Fingerprint Image Processing

In order for the RaspiReader to be used for spoof detection, it must also demonstrate the ability to output high quality fingerprint images suitable for fingerprint matching. As previously mentioned, the RaspiReader performs spoof detection on non-processed, raw fingerprint images. While these raw images are shown to provide discriminatory information for spoof detection, they need to be made compatible with processed images output by other COTS fingerprint readers. Therefore, after spoof detection, the RaspiReader performs image processing operations on the raw high contrast, FTIR image frames in order to output high fidelity images compatible with COTS optical fingerprint readers.

Let a raw (unprocessed) FTIR fingerprint image from the RaspiReader be denoted as $FTIR_{raw}$. This raw image $FTIR_{raw}$ is first converted from the RGB color space to grayscale ($FTIR_{gray}$) (Fig. 3.6 (a)). Then, in order to further contrast the ridges from the valleys of the fingerprint, histogram equalization is performed on $FTIR_{gray}$ (Fig. 3.6 (b)). Finally, $FTIR_{gray}$ is negated so

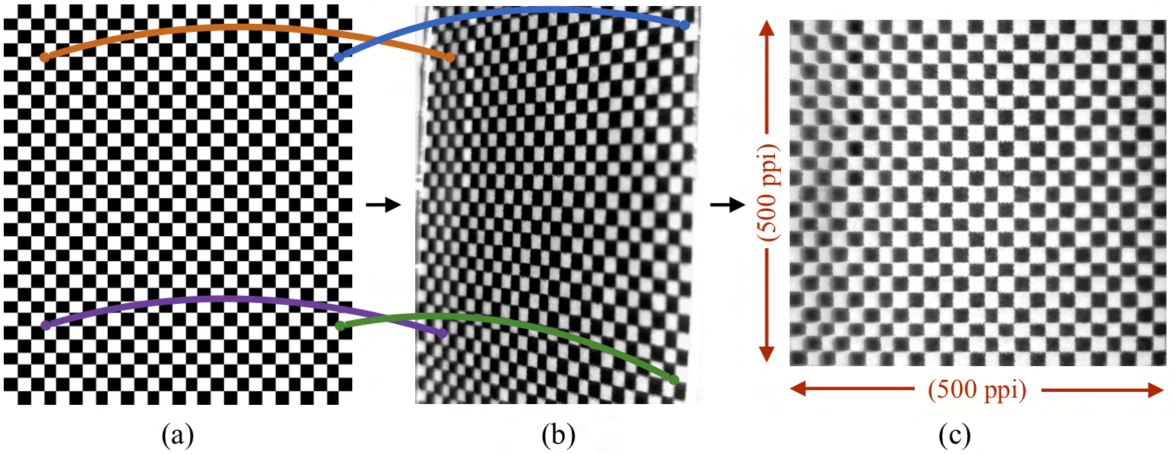


Figure 3.7 Acquiring Image Transformation Parameters. A 2D printed checkerboard pattern (a) is imaged by the RaspiReader (b). Corresponding points between the frontalized checkerboard pattern (a) and the distorted checkerboard pattern (b) are defined so that perspective transformation parameters can be estimated to map (b) into (c). These transformation parameters are subsequently used to frontalize fingerprint images acquired by RaspiReader for the purpose of fingerprint matching. The checkerboard imaged in (b) is also used to acquire the native resolution of RaspiReader in order to scale matching images to 500 ppi in both the x and y axis as shown in (c).

that the ridges of the fingerprint image are dark, and the background of the image is white (as are fingerprint images acquired from COTS readers) (Fig. 3.6 (c)).

Following the aforementioned image processing techniques, the RaspiReader FTIR fingerprint images are further processed by performing a perspective transformation (to frontalize the fingerprint to the image plane) and scaling to 500 ppi (Figs. 3.6 (d), (f)). Note, we also experimented with non-linear distortion corrections (camera barrel distortion), but found no improvement (over a simple linear distortion correction) in RaspiReader matching performance and little improvement in error between landmarks on the ground truth calibration pattern and the non-linear distortion corrected images. This makes sense, since the distortion of the raw FTIR images can be seen to be predominantly tangential (Fig. 3.7 (b)).

A perspective transformation is performed using Equation 3.2.2,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.2.2}$$

where x and y are the source coordinates, x' and y' are the transformed coordinates, (a, b, c, d, e, f, g, h) is the set of transformation parameters, and $\lambda = gx + hy + 1$ is a scale parameter. In this work, we image a 2D printed checkerboard pattern[6] to define source and destination coordinate pairs such that the transformation parameters could be estimated (Fig. 3.7). Once the perspective transformation has been completed, the RaspiReader image is downsampled (by averaging neighborhood pixels) to 500 ppi (Fig. 3.6 (f)). Note that the native resolution of the RaspiReader images was acquired using a 2D printed checkerboard calibration pattern (Fig. 3.7 (b)) and ranges from approx. 1594 ppi to 2480 ppi in the x-axis (Fig. 3.8 (a)) and 2463 ppi to 3320 ppi in the y-axis (Fig. 3.8 (b)). While the high resolution images captured by the RaspiReader 5 Megapixel cameras far exceed the resolution of COTS fingerprint readers (providing added minute textural details for distinguishing live fingers from spoof fingers), we observed that the focus of the native images captured by RaspiReader does deteriorate on the left and right edges (Fig. 3.7 (b)). We are currently investigating methods for properly focusing the lens on the entire FOV, so that minute textural details are not lost on the edges of the RaspiReader images.

Upon completion of this entire fingerprint reader assembly and image processing procedure, the RaspiReader is fully functional and ready for use in both spoof detection and subsequent fingerprint matching.

[6] A checkerboard can be imaged by RaspiReader by printing a checkerboard pattern on glossy paper and applying several drops of water to the platen prior to placing the printed checkerboard.

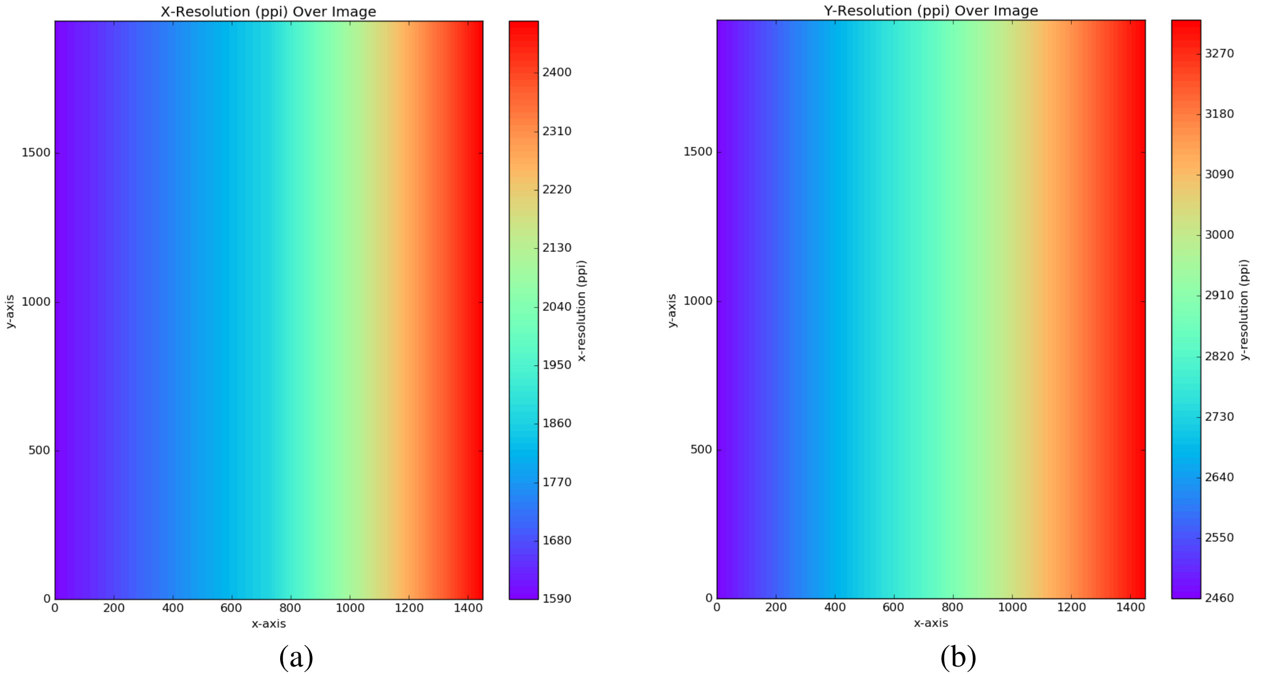


Figure 3.8 Native resolution (ppi) in (a) x-axis and (b) y-axis over the raw FTIR RaspiReader image. As is normal, native resolution changes across the image because the right side of the image is closer to the camera than the left side.

3.3 Live and Spoof Fingerprint Database Construction

To test the utility of the RaspiReader for spoof detection and its interoperability for fingerprint matching, a database of live and spoof fingerprint impressions was collected for performing experiments. This database is constructed as follows.

Using 7 different materials (Fig. 3.3 (a)), 66 spoofs were fabricated[7]. Then, for each of these spoofs, 10 impressions were captured at varying orientations and pressure on both the RaspiReader (Rpi) and a COTS 500 ppi, PIV-certified, optical FTIR fingerprint reader ($COTS_A$). Note, we also experimented with an Appendix-F certified slap scanner but found little difference in spoof detection performance between the PIV-certified device and the Appendix-F certified device. The summary of this data collection is enumerated in Table 3.1.

[7] Our spoofs were shipped to us by Precise Biometrics [139], a company specializing in evaluating spoof detection capability and that also has close ties to the LivDet dataset authors. As such, our spoofs are of high quality and are similar to the spoofs used in the LivDet competition.

Table 3.2 Summary of Spoof Fingerprints Collected

Material[1]	Spoof Count[2]	RPi Direct Images	RPi FTIR Images	$COTS_A$ FTIR Images
Ecoflex	10	100	100	100
Wood Glue	10	100	100	100
Monster Liquid Latex	10	100	100	100
Liquid Latex Body Paint	10	100	100	100
Gelatin	10	100	100	100
Silver Coated Ecoflex	10	100	100	100
Crayola Model Magic	6	60	60	60
Total	66	660	660	660

[1] The spoof materials used to fabricate these spoofs were in accordance with the approved materials by the IARPA ODIN project [76].
[2] The spoofs are all of unique fingerprint patterns.

To collect a sufficient variety of live finger data, we enlisted 15 human subjects with different skin colors (Fig. 3.3 (b)). Each of these subjects gave 5 finger impressions (at different orientations and pressures) from all 10 of their fingers on both the RaspiReader and $COTS_A$. A summary of this data collection is enumerated in Table 3.2.

Table 3.3 Summary of Live Finger Data Collected

Number of Subjects	Number of Fingers	RPi Direct Images	RPi FTIR Images	$COTS_A$ FTIR Images
15	150	750	750	750

In addition to the images of live finger impressions and spoof finger impressions we collected for conducting spoof detection experiments, we also verified that for spoofs with optical properties too far from that of live finger skin (Fig. 3.9), images would not be captured by the RaspiReader.

These "failure to capture" spoofs are therefore filtered out as attacks before any software based spoof detection methods need to be performed.

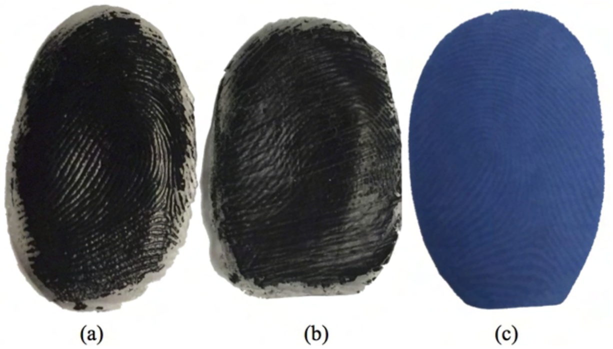


Figure 3.9 Failure to Capture. Several spoofs are unable to be imaged by the RaspiReader due to their dissimilarity in color. In particular, because spoofs in (a) and (b) are black, all light rays will be absorbed preventing light rays from reflecting back to the FTIR imaging sensor. In (c), the dark blue color again prevents enough light from reflecting back to the camera. (a) and (b) are both ecoflex spoofs coated with two different conductive coatings. (c) is a blue crayola model magic spoof attack.

3.4 Spoof Detection Experiments and Results

Given the database of live and spoof fingerprint images collected on both $COTS_A$, and the prototype RaspiReader, a number of spoof detection experiments are conducted to demonstrate the superiority of the raw images from the RaspiReader for training spoof detectors in comparison to the grayscale images output by COTS optical readers. In particular, we (i) take several successful spoof detection techniques from the literature, (ii) train and test the spoof detectors on $COTS_A$ images, (iii) train and test the spoof detectors on RaspiReader images, and (iv) compare the results to show the significant boost in performance when RaspiReader images are used to train spoof detectors rather than $COTS_A$ images. In addition, experiments are conducted to demonstrate that fingerprint images from the RaspiReader are compatible for matching with fingerprint images acquired from $COTS_A$.

3.4.1 Spoof Detection Methods

To thoroughly demonstrate the value RaspiReader images provide in training spoof detectors, we select two different spoof detection methods, namely, (i) textural features (LBP [132]) in conjunction with a linear Support Vector Machine (SVM) and (ii) a Convolutional Neural Network (CNN). Textural features were chosen because of their popularity and their demonstrated superior spoof detection performance in comparison to other "hand-crafted features" such as anatomical or physiological features in the literature [185]. CNNs were chosen as a second spoof detection method in our experiments given that they are currently state-of-the-art on the publicly available LivDet datasets. [26, 117, 131, 132]. The details of the experiments performed with both of these spoof detection methods are provided in the following subsections.

3.4.1.1 LBP Features From $COTS_A$ Images

We begin our experiments using grayscale processed fingerprint images acquired from $COTS_A$ (Fig. 3.2 (c)). From these images, we extract the very prevalent grayscale and rotation invariant local binary patterns (LBP) [132]. LBP features are extracted by constructing a histogram of bit string values determined by thresholding pixels in the local neighborhoods around each pixel in the image. Since image texture can be observed at different spatial resolutions, parameters R and P are specified in LBP construction to indicate the length (in pixels) of the neighborhood radius used for selecting pixels and also the number of neighbors to consider in a local neighborhood. Previous studies have shown that more than 90% of fundamental textures in an image can belong to a small subset of binary patterns called "uniform" textures (local binary patterns containing two or fewer 0/1 bit transitions) [132]. Therefore, in line with previous studies using local binary patterns for fingerprint spoof detection, we also employ the use of uniform local binary patterns.

More formally, let $LBP(P, R)$ be the uniform local binary pattern histogram constructed by binning the local binary patterns for each pixel in an image according to the well known LBP operation [132] with parameters P and R. In our experiments, we extract $LBP(8, 1)$, $LBP(16, 2)$, and $LBP(24, 3)$ in order to capture textures at different spatial resolutions. These histograms (each having $P + 2$ bins) are individually normalized and concatenated into a single feature vector $\mathbf{X}$ of dimension 54.

For classification of these features, we employ a binary linear SVM. As known in the art, an initially "hard margin" SVM can be "softened" by a parameter C to enable better generalization of the classifier to the testing dataset. In our case, we use five-fold cross validation to select the value of C (from the list of $\begin{bmatrix} 10^{-5} & 10^{-4} & ... & 10^{4} & 10^{5} \end{bmatrix}$) such that the best performance is achieved in different folds. In our experiments, the best classification results were achieved with $C = 10^2$.

3.4.1.2 CLBP Features From RaspiReader Images

In this experiment, we make use of the information rich images from the RaspiReader (Figs. 3.2 (a, b)) for spoof detection. As with Experiment 1, we again pursue the use of LBP textural features. However, since the raw images from the RaspiReader contain color information, rather than using the traditional grayscale LBP features, we employ the use of color local binary patterns (CLBP). Previous works have shown the efficacy of CLBP for both face recognition and face spoof detection [15,25]. However, because fingerprint images from COTS fingerprint readers are grayscale, CLBP features have, to our knowledge, not been investigated for use in fingerprint spoof detection until now.

Unlike traditional grayscale LBP patterns, color local binary patterns (CLBP) encode discriminative spatiochromatic textures from across multiple spectral channels [25]. In other words, CLBP extracts textures across all the different image bands in a given input image. More formally, given an input image I with K spectral channels, let the set of all spectral channels for I be defined as $S = \{S_1, ..., S_K\}$. Then, the CLBP feature vector $\mathbf{X}$ of dimension 486 can be extracted from I using Algorithm 1. Note that in Algorithm 1, $LBP(S_i, S_j, P, R)$ returns a normalized histogram of local binary patterns using S_i as the image channel that the center (thresholding) pixels are selected from, and S_j as the image channel from which the neighborhood pixels are selected from in the same computation of LBP as performed in Experiment 1. Also note that in Algorithm 1, $\|$ indicates vector concatenation. Finally, in our experiments, we preprocess the RaspiReader input image I prior to CLBP extraction by (i) downsampling (FTIR images from 1450 x 1944 to 108 x 145 and direct view images from 1290 x 1944 to 96 x 145), and (ii) converting to the HSV color space[8].

Algorithm 1 Extraction of Color Local Binary Patterns

```
$\mathbf{X} \leftarrow [\,]$
**for** $i \leftarrow 1, K$ **do**
    **for** $j \leftarrow 1, K$ **do**
        $\mathbf{X} \leftarrow \mathbf{X} \| LBP(S_i, S_j, 8, 1) \|$
            $LBP(S_i, S_j, 16, 2) \| LBP(S_i, S_j, 24, 3)$
    **end for**
**end for**
**return** $\mathbf{X}$
```

As in Experiment 1, a binary linear SVM with a parameter of $C = 10^2$ is trained with these features and subsequently used for classification. We again choose the parameter C using 5-fold cross validation and a selection list of $\begin{bmatrix} 10^{-5} & 10^{-4} & ... & 10^{4} & 10^{5} \end{bmatrix}$. Since RaspiReader outputs two color images (one raw FTIR image and one direct view image), we perform multiple experiments using the proposed CLBP features in conjunction with the SVM. In particular, we (i) extract CLBP features from the RaspiReader raw FTIR images to train/test a SVM, (ii) extract CLBP features from RaspiReader direct view images to train/test a SVM, and (iii) fuse CLBP features from both image outputs to train and test a SVM. We also attempted fusing CLBP features from the RaspiReader raw images with grayscale LBP features from RaspiReader processed FTIR images, but found no significant performance gains under this last fusion scheme.

[8] Other color spaces were experimented with, but HSV consistently provided the highest performance. This is likely because HSV separates the luminance and chrominance components in an image, allowing extraction of features on more complementary image channels.

3.4.1.3 MobileNet

In addition to performing experiments involving "handcrafted" textural features, we also perform experiments where the features are directly learned and classified by a Convolutional Neural Network (CNN). In choosing a CNN architecture, we carefully considered both the size and computational overhead, since in future works, we will optimize the architecture to directly run on the RaspiReader's Raspberry Pi Processor. The need for a "low over-head" architecture prompted us to select MobileNet [75]. MobileNet has been shown to perform very closely (within 1 % accuracy) to popular CNN models (VGG and Inception v3) on the ImageNet and Stanford Dogs datasets while being 32 times smaller than VGG, 7 times smaller than Inception v3, 27 times less computationally expensive than VGG, and 8 times less computationally expensive than Inception v3.

In our experiments, we employ the Tensorflow Slim implementation of MobileNet[9]. MobileNet is comprised of 28 convolutional layers, and in our case, a final 2 class softmax layer for classification of live or spoof. In all of our experiments involving MobileNet, the RMSProp optimizer was used for training the network along with a batch size of 32, and an adaptive (exponential decay) learning rate. To increase the generalization ability of the networks, we employ various data augmentation methods such as brightness adjustment, random cropping, and horizontal and vertical reflections.

Using the aforementioned MobileNet architecture and hyper-parameters, we train/test the network with (i) $COTS_A$ grayscale fingerprint images, (ii) RaspiReader raw FTIR images, (iii) RaspiReader direct view images, and (iv) RaspiReader processed FTIR images. Additionally, we perform experiments in which we fuse the score outputs of MobileNet models trained on the different image outputs from RaspiReader to take advantage of the complementary information within the different RaspiReader image outputs. When training and testing MobileNet with $COTS_A$ images or RaspiReader processed FTIR images, the three input channels of the network are each fed with the same down sampled (357 x 392 to 224 x 224) grayscale $COTS_A$ image or (290 x 267

[9] https://github.com/tensorflow/models/tree/master/research/slim

to 224 x 224) RaspiReader processed FTIR image. When training the network with RaspiReader raw images, we again down sample the images (1450 x 1944 to 224 x 224 for raw FTIR and 1290 x 1944 to 224 x 224 for direct image), however, in this case, each of the three color channels are fed as input to the three input channels of the network. More specifically, we first convert the RaspiReader image to HSV (given our earlier findings of superior performance in this color space), and then feed each channel H, S, and V into the network's input channels.

3.4.2 Spoof Detection Results

Using the spoof detection schemas previously described, we train and test classifiers under two main scenarios. In the first scenario, we train the classifier on a subset of spoof images from every type in the dataset (Table 3.1). During testing, spoof images from the same spoof types seen during training will be passed to the spoof detector for classification. We hereafter refer to this training and testing scenario as a "known-material" scenario. In the second scenario, we train the classifier with images from all of the spoof types in the dataset except one (i.e. the spoof impressions from one type of spoof are withheld). Then, during testing the impressions of the withheld spoof type are used for testing. In the literature, this type of spoof detection evaluation is referred to as a "cross-material" scenario. In the following experimental results, we demonstrate that the RaspiReader images significantly boost the spoof detection performance in both the known-material evaluations and the cross-material evaluations.

3.4.2.1 Known-Material Scenarios

The first known-material results are reported in accordance with spoof detection methods 1 and 2. That is, we extract textural features from both $COTS_A$ images and RaspiReader images respectively, train and test linear SVMs, and finally, compare the results (Table 3.3). In all of our known-material scenario experiments, we report the average spoof detection performance and standard deviation over 5-folds. That is, for spoof data, we select 80% of the spoof impressions from each spoof material for training (each fold) and use the remaining 20% for testing. For live finger data, we select the finger impressions of 12 subjects each fold (600 total images) for training, and use the live finger impressions of the remaining 3 subjects for testing.

Table 3.4 Textural Features and Known Testing Materials

Method	TDR @ FDR = 1.0% $\mu \pm \sigma^1$	Detection Time (msecs)
$COTS_A$ + LBP	$75.9\% \pm 30.8$	236
Rpi raw FTIR + CLBP	$91.5\% \pm 11.0$	243
Rpi Direct + CLBP	$98.10\% \pm 1.9$	243
Rpi Fusion + CLBP[2]	$97.7\% \pm 3.0$	486

[1] These results are reported over 5-folds.

[2] Rpi Fusion + CLBP is a feature level fusion (concatenation) of CLBP features extracted from both Rpi raw FTIR images and Rpi Direct Images, respectively.

From the results of Table 3.3, one can observe that both image outputs of the RaspiReader contain far more discriminative information for spoof detection than the processed grayscale images output by $COTS_A$. In particular, spoof detection performance is significantly higher when extracting textural (CLBP) features from the RaspiReader images, than when extracting textural features (LBP) from $COTS_A$ images. While in these first results, the fusion of features from both RaspiReader image outputs actually hurts the classification performance slightly (compared to extracting features only from the direct view images), in subsequent experiments, we will demonstrate that different feature extraction and classification techniques can better utilize the multiple outputs of RaspiReader in a complementary manner to instead boost the classification performance.

The second known-material results are reported in accordance with spoof detection scheme 3. More specifically, the results are reported (over 5-folds) when MobileNet is trained and tested with (i) $COTS_A$ images, (ii) RaspiReader processed FTIR images, (iii) RaspiReader raw FTIR images, and (iv) RaspiReader direct images. In addition, we report the results when fusing the score outputs of multiple MobileNet models trained on the different image outputs of RaspiReader (Table 3.4).

Table 3.5 MobileNet and Known Testing Materials

Method	TDR @ FDR = 1.0% $\mu \pm \sigma$[1]	Detection Time (msecs)
$COTS_A$ + MobileNet	$91.9\% \pm 8.0$	22
Rpi processed FTIR + MobileNet	$94.5\% \pm 3.7$	22
Rpi raw FTIR + MobileNet	$95.1\% \pm 5.6$	22
Rpi Direct + MobileNet	$95.3\% \pm 3.5$	22
Rpi Fusion 2 + MobileNet[2]	$98.4\% \pm 2.3$	45
Rpi Fusion 3 + MobileNet[3]	$98.9\% \pm 1.5$	67

[1] These results are reported over 5-folds.
[2] Rpi Fusion 2 + MobileNet is a score level fusion (averaging) of a MobileNet model trained on Rpi raw FTIR images and a MobileNet model trained on Rpi Direct Images.
[3] Rpi Fusion 3 + MobileNet is a score level fusion (averaging) of separate MobileNet models trained on Rpi raw FTIR images, Rpi Direct Images, and on Rpi processed FTIR images.

The results of Table 3.4 show that both the raw image outputs of RaspiReader and the processed image output of RaspiReader contain more discriminative information for spoof detection than the processed images output by $COTS_A$. The MobileNet models trained on RaspiReader images always outperform the MobileNet model trained on $COTS_A$ grayscale images both in average spoof detection performance and stability (significantly lower s.d.). What is further interesting about the results of Table 3.4 is that the features extracted by MobileNet from each RaspiReader output are quite complementary, demonstrated by the fact that spoof detection performance is improved when fusing the scores of MobileNet models trained on each RaspiReader image output. So, while CLBP features outperform MobileNet on the RaspiReader direct images, the fused MobileNet classifiers outperform the fused CLBP classifier.

3.4.2.2 Cross-Material Scenarios

The cross-material results use the same spoof detection schemas as enumerated in the known-material results with a primary difference being the training and testing data splits provided to

Table 3.6 Textural Features and Cross-Material Testing[1]

Testing Material	$COTS_A$ + LBP	Rpi Fusion + CLBP[2]
Crayola Model Magic	91.7%	98.3%
Ecoflex	66.0%	77.0%
Silver Coated Ecoflex	88.0%	100.0%
Gelatin	62.0%	87.0%
Liquid Latex Body Paint	84.0%	100.0%
Monster Liquid Latex	68.0%	98.0%
Wood Glue	100.0%	81.0%

[1] TDR @ FDR = 1.0% is reported

[2] Rpi Fusion + CLBP is a feature level fusion (concatentation) of CLBP features extracted from both Rpi raw FTIR images and Rpi Direct Images, respectively.

the various classifiers. In all the cross-material scenarios, spoof impressions of six materials are partitioned to the classifier for training, and the spoof impressions of one "unseen" material are kept aside for testing. In this manner the generalization capability of the spoof detector to novel spoof types is thoroughly assessed. For live finger data, we randomly select the finger impressions of two subjects (100 total images) for testing, and use the live finger impressions of the remaining thirteen subjects for training. Since there are seven different spoof materials in our training set (Table 3.1), we conduct seven different cross-material experiments for each spoof detection schema (where one of the seven spoof types is left aside for testing). The cross material results when using textural features in conjunction with SVMs is reported in Table 3.5. The cross-material results when using MobileNet extracted features is reported in Table 3.6. Finally, we report the spoof detection accuracy of a state-of-the-art, commercial fingerprint reader (with embedded spoof detection capabilities) (Lumidigm V-Series [73]) to further provide a fair comparison of our proposed RaspiReader to current state-of-the-art hardware based spoof detection systems (Table **??**). Note, we only report the best textural fusion and CNN fusion methods in the cross-material results. The

Table 3.7 MobileNet and Cross-Material Testing[1]

Testing Material	$COTS_A$ + MobileNet	Rpi Fusion 2 + MobileNet[2]	Rpi Fusion 3 + MobileNet[3]
Crayola Model Magic	50.0%	100.0%	100.0%
Ecoflex	100.0%	8.0%	56.0%
Silver Coated Ecoflex	77.0%	100.0%	100.0%
Gelatin	88.0%	100.0%	100.0%
Liquid Latex Body Paint	97.0%	100.0%	100.0%
Monster Liquid Latex	86.0%	100.0%	100.0%
Wood Glue	94.0%	96.0%	96.0%

[1] TDR @ FDR = 1.0% is reported.
[2] Rpi Fusion 2 + MobileNet is a score level fusion (max) of separate MobileNet models trained on Rpi raw FTIR images and on Rpi Direct Images, respectively.
[3] Rpi Fusion 3 + MobileNet is a score level fusion (max) of separate MobileNet models trained on Rpi raw FTIR images, Rpi Direct Images, and on Rpi processed FTIR images.

Table 3.8 Lumidigm Spoof Detection Accuracy

Crayola Model Magic	Ecoflex	Silver Coated Ecoflex	Gelatin	Liquid Latex Body Paint	Monster Liquid Latex	Wood Glue
100%	100%	100%	90%	100%	50%	45%

[1] Lumidigm classification accuracy is reported to enable further comparison of RaspiReader to state-of-the-art spoof resistant fingerprint readers. [2] For each material, the same spoofs used to construct Table 2 were imaged on Lumidigm (20 impressions per material).

other non-fusion based methods were experimented with, but did not provide as high of performance in the cross-material scenarios.

The key findings of the cross-material experiments as revealed in Tables 3.5, 3.6, and 3.7 are as follows. First, in both textural based spoof detection methods and CNN based spoof detection methods, the raw images output by RaspiReader almost always provide more discriminative information than COTS grayscale fingerprint images. This enables much higher spoof detection performance on spoofs fabricated from materials not seen by the classifier during training, a major flaw in many existing spoof detection methods relying on only COTS grayscale images. We also note that the RaspiReader cross-material performance is significantly higher than Lumidigm (Tables 3.6 and 3.7) on several spoof materials, further demonstrating RaspiReader's effectiveness in comparison to current state-of-the-art, commercial, hardware based spoof detection techniques.

The one case of poor cross-material performance (when using RaspiReader images) came when the testing material withheld was ecoflex (Table 3.6). This can be explained by ecoflex being a very transparent spoof, enabling much of the live finger color behind the spoof to seep through. As such, when the MobileNet models were trained on the other non-transparent spoofs and tested on the transparent ecoflex, the performance dropped considerably. However, we also noticed that the best cross-material performance (when using $COTS_A$ images) came when the testing material withheld was ecoflex. The most plausible explanation for this is that the MobileNet model trained on the $COTS_A$ images must focus on textural features rather than color. As such, the transparent property of ecoflex did not affect the classifier trained on the grayscale images. This prompted us to train a third model on the RaspiReader processed FTIR images (i.e. the raw FTIR images were converted to grayscale and contrast enhanced). We then fused the score of this third model with the two MobileNet models trained on the RaspiReader raw FTIR and direct images respectively. The final product was a three CNN model system which performed much better on the ecoflex testing scenario (48$\%$ improvement). Note, the standalone performance of the MobileNet model trained on RaspiReader processed FTIR images was 86.0%, lending evidence to our hypothesis that RaspiReader performed worse on ecoflex due to the transparent nature of the material. While the ecoflex testing scenario is still low under fusion (56.0$\%$), in a real world setting, this limitation is easily solved by including one transparent spoof in the training set (evidenced by the fact that in the known-material experiments, ecoflex could be differentiated from live fingers with high accuracy).

We also note that in Table 3.5, $COTS_A$ outperformed RaspiReader when the testing material withheld was wood glue. Again, in this specific testing scenario, the grayscale features were better able to generalize to the unknown spoof material. This is evidenced by the fact that when using grayscale LBP features extracted from the RaspiReader matching images, the TDR was computed to be 93.0%, a significant improvement from the CLBP performance of 81.0%. The remaining difference between the RaspiReader grayscale LBP performance on wood glue and $COTS_A$ performance on wood glue can likely be attributed to subtle differences in the image processing techniques used by the two readers in converting a raw FTIR image to a processed FTIR image.

3.5 Interoperability of RaspiReader

In addition to demonstrating the usefulness of the RaspiReader images for fingerprint spoof detection, we also demonstrate that by processing the RaspiReader FTIR images, we can output images which are compatible for matching with images from COTS fingerprint readers. Previously, we discussed how to process and transform a RaspiReader raw FTIR image into an image suitable for matching. In this experiment, we evaluate the matching performance (of 11,175 imposter pairs and 6,750 genuine pairs) when using (i) the RaspiReader processed images as both the enrollment and probe images, (ii) the $COTS_A$ images as both the enrollment and probe images, (iii) the $COTS_A$ images as the enrollment images and the RaspiReader processed images as the probe images, and (iv) the RaspiReader images as the enrollment images and the $COTS_A$ images as the probe images. The results for these matching experiments are listed in Table 3.8.

From these results, we make two observations. First, the best performance is achieved for native comparisons, where the enrolled and search (probe) images are produced by the same capture device. RaspiReader's native performance is slightly better than that of $COTS_A$. This indicates that the RaspiReader is capable of outputting images which are compatible with state of the art fingerprint matchers. Second, we note that the performance does drop slightly when conducting

Table 3.9 Fingerprint Matching Results

Enrollment Reader	Probe Reader	TAR @ FAR = 0.1%[†]
$COTS_A$	$COTS_A$	98.62%
RaspiReader	RaspiReader	99.21%
$COTS_A$	RaspiReader	95.56%
RaspiReader	$COTS_A$	95.10%

[†] We use the Innovatrics fingerprint SDK which is shown to have high accuracy in the NIST FpVTE evaluation [178].

the interoperability experiment ($COTS_A$ is used for enrollment images and RaspiReader is used for probe images). However, the matching performance is still quite high considering the stringent operating point (FAR = 0.1%). Furthermore, studies have shown that when different fingerprint readers are used for enrollment and subsequent verification or identification, the matching performance indeed drops [43, 94, 144]. Finally, we are currently investigating other approaches for processing and downsampling RaspiReader images to reduce some of the drop in cross-reader performance.

3.6 Computational Resources

All image preprocessing, LBP and CLBP feature extractions, and SVM classifications were performed with a single CPU core on a Macbook Pro running a 2.9 GHz Intel Core i5 processor. MobileNet training and classification was performed on a single Nvidia GTX Titan GPU. The total time from image capture to spoof detection with our best MobileNet model (RpiFusion3) is approximately 3.067 seconds (1.5 seconds for image capture, 1.5 seconds to transmit data to GPU, and 67 milliseconds for classification).

3.7 Summary

We have open sourced[10] the design and assembly of a custom fingerprint reader, called RaspiReader, with Raspberry Pi and other ubiquitous components. This fingerprint reader is both low cost (US $175) and easy to assemble[11], enabling other researchers to easily and seamlessly develop their own novel fingerprint spoof detection solutions which use both hardware and software. By customizing RaspiReader with two cameras for fingerprint image acquisition rather than the customary one, we were able to extract discriminative information from both raw images which, when fused together, enabled us to achieve higher spoof detection performance (in both known-material and cross-material testing scenarios) compared to when features were extracted from COTS grayscale images. Finally, by processing the raw FTIR images of the RaspiReader, we were able to output fingerprint images compatible for matching with COTS optical fingerprint readers demonstrating the interoperability of RaspiReader.

Future directions could be to integrate specialized hardware into RaspiReader such as IR cameras for vein detection, or microscopes for capturing extremely high resolution images of the fingerprint. Because the RaspiReader uses ubiquitous components running open source software, RaspiReader enables integration of these additional hardware components. In addition to the integration of specialized hardware, one could also pursue use of the raw, information rich images from the RaspiReader to pursue one-class classification schemes for fingerprint spoof detection such as that proposed in [45].

3.8 Acknowledgment

This research was supported by grant no. 70NANB17H027 from the NIST Measurement Science program and 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

[10] https://github.com/engelsjo/RaspiReader

[11] https://bit.do/RaspiReader

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.

We would like to thank Chris Perry for all of his help and support as a system integrator throughout this project.

Chapter 4

Learning a Fixed Length Fingerprint Representation

In the previous chapters, our focus was primarily on improving the performance and security of the fingerprint *reader* sub-module of fingerprint recognition systems via Universal Target evaluations and the spoof resistant RaspiReader, respectively. In this chapter, we turn our focus towards improving the performance (accuracy and speed) and security of the fingerprint *feature extraction* and *matching* sub-modules of fingerprint recognition systems. In particular, we present DeepPrint, a deep network, which learns to extract fixed-length fingerprint representations of only 200 bytes [48]. The DeepPrint representation enables accuracy levels comparable to state-of-the-art AFIS based upon minutiae representations and can be matched at orders of magnitude faster speeds. Furthermore, the DeepPrint representation can be more easily secured and matched in the encrypted domain via a fully homomorphic encryption scheme.

To arrive at a discriminative fixed-length representation, DeepPrint incorporates fingerprint domain knowledge, including alignment and minutiae detection, into a deep network architecture. We benchmark DeepPrint against two top performing COTS SDKs (Verifinger and Innovatrics) from the NIST and FVC evaluations. Coupled with a re-ranking scheme, the DeepPrint rank-1 search accuracy on the NIST SD4 dataset against a gallery of 1.1 million fingerprints is comparable to the top COTS matcher, but it is significantly faster (**DeepPrint:** 98.80% in 0.3 seconds vs. **COTS A:** 98.85% in 27 seconds). To the best of our knowledge, the DeepPrint representation is the most compact and discriminative fixed-length fingerprint representation reported in the academic literature.

4.1 Introduction

To overcome the limitations of the variable length minutiae representation (Table 4.1), we present a reformulation of the fingerprint recognition problem. In particular, rather than extracting varying length minutiae-sets for matching (*i.e.* handcrafted features), we design a deep network embedded with fingerprint domain knowledge, called **DeepPrint**, to *learn* a fixed-length representation of 200 bytes which discriminates between fingerprint images from different fingers (Fig. 4.1).

Table 4.1 Comparison of variable length minutiae representation with fixed-length DeepPrint representation

Matcher	(Min, Max) # of Minutiae[1]	(Min, Max) Template Size (kB)
COTS A	(12, 196)	(1.5, 23.7)
COTS B	(12, 225)	(0.6, 5.3)
Proposed	N.A.[2]	0.2[†]

[1] Statistics from NIST SD4 and FVC 2004 DB1.
[2] Template is not explicitly comprised of minutiae.
[†] Template size is fixed at 200 bytes, irrespective of the number of minutiae (192 bytes for the features and 8 bytes for 2 decompression scalars).

To arrive at a compact and discriminative representation of only 200 bytes, the DeepPrint architecture is embedded with fingerprint domain knowledge via an automatic alignment module and a multi-task learning objective which requires minutiae-detection (in the form of a *minutiae-map*) as a side task to representation learning. More specifically, DeepPrint automatically aligns an input fingerprint and subsequently extracts both a *texture representation* and a *minutiae-based representation* (both with 96 features). The 192-dimensional concatenation of these two representations, followed by compression from floating point features to integer value features comprises

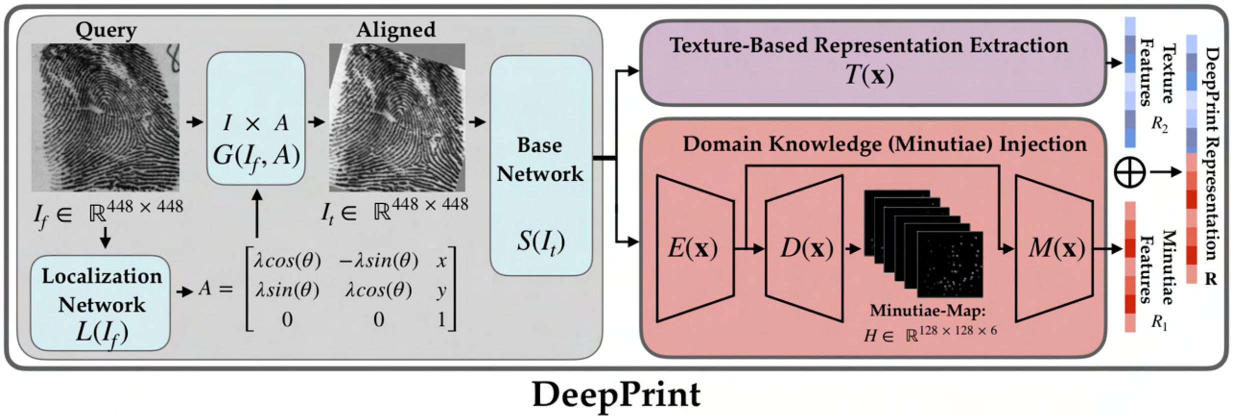


Figure 4.1 Flow diagram of DeepPrint: (i) a query fingerprint is aligned via a Localization Network which has been trained end-to-end with the Base-Network and Feature Extraction Networks (no reference points are needed for alignment); (ii) the aligned fingerprint proceeds to the Base-Network which is followed by two branches; (iii) the first branch extracts a 96-dimensional texture-based representation; (iv) the second branch extracts a 96-dimensional minutiae-based representation, guided by a side-task of minutiae detection (via a minutiae map which does not have to be extracted during testing); (v) the texture-based representation and minutiae-based representation are concatenated into a 192-dimensional representation of 768 bytes (192 features and 4 bytes per float). The 768 byte template is compressed into a 200 byte fixed-length representation by truncating floating point value features into integer value features, and saving the scaling and shifting values (8 bytes) used to truncate from floating point values to integers. The 200 byte DeepPrint representations can be used both for authentication and large-scale fingerprint search. The minutiae-map can be used to further improve system accuracy and interpretability by re-ranking candidates retrieved by the fixed-length representation.

a 200 byte fixed-length representation (192 bytes for the feature vector and 4 bytes for storing the 2 compression parameters). As a final step, we utilize Product Quantization [93] to further compress the DeepPrint representations stored in the gallery, significantly reducing the computational requirements and time for large-scale fingerprint search.

Detecting minutiae (in the form of a minutiae-map) as a side-task to representation learning has several key benefits:

- We guide our representation to incorporate domain inspired features pertaining to minutiae by sharing parameters between the minutiae-map output task and the representation learning task in the multi-task learning framework.

- Since minutiae representations are the most popular for fingerprint recognition, we posit that our method for guiding the DeepPrint feature extraction via its minutiae-map side-task falls in line with the goal of "Explainable AI" [34].

- Given a probe fingerprint, we first use its DeepPrint representation to find the top k candidates and then re-rank the top k candidates using the minutiae-map provided by DeepPrint [1]. This *optional* re-ranking add-on further improves both accuracy and interpretability.

The primary benefit of the 200 byte representation extracted by DeepPrint comes into play when performing mega-scale search against millions or even billions of identities (e.g., India's Aadhaar [171] and the FBI's Next Generation Identification (NGI) databases [56]). To highlight the significance of this benefit, we benchmark the search performance of DeepPrint against the latest version SDKs (as of July, 2019) of two top performers in the NIST FpVTE 2012 (Innovatrics[2] v7.2.1.40 and Verifinger[3] v10.0[4]) on the NIST SD4 [129] and NIST SD14 [128] databases augmented with a gallery of nearly 1.1 million rolled fingerprints. Our empirical results demonstrate that DeepPrint is competitive with these two state-of-the-art COTS matchers in accuracy while requiring only a fraction of the search time. Furthermore, a given DeepPrint fixed-length representation can also be matched in the encrypted domain via homomorphic encryption with minor loss to recognition accuracy as shown in [14] for face recognition.

More concisely, the primary contributions of this chapter are:

- A customized deep network (Fig. 4.1), called DeepPrint, which utilizes fingerprint domain knowledge (alignment and minutiae detection) to learn and extract a discriminative fixed-length fingerprint representation.

- Demonstrating in a manner similar to [177] that Product Quantization can be used to compress DeepPrint *fingerprint* representations, enabling even faster mega-scale search (51 ms

[1] The $128 \times 128 \times 6$ DeepPrint minutiae-map can be easily converted into a minutiae-set with n minutia: $\{(x_1, y_1, \theta_1), ..., (x_n, y_n, \theta_n)\}$ and passed to any minutia-matcher (e.g., COTS A, COTS B, or [20]).

[2] https://www.innovatrics.com/

[3] https://www.neurotechnology.com/

[4] We note that Verifinger v10.0 performs significantly better than earlier versions of the SDK often used in the literature.

search time against a gallery of 1.1 million fingerprints vs. 27,000 ms for a COTS with comparable accuracy).

- Demonstrating with a two-stage search scheme similar to [177] that candidates retrieved by DeepPrint representations can be re-ranked using a minutiae-matcher in conjunction with the DeepPrint minutiae-map. This further improves system interpretability and accuracy and demonstrates that the DeepPrint features are complementary to the traditional minutiae representation.

- Benchmarking DeepPrint against two state-of-the-art COTS matchers (Innovatrics and Verifinger) on NIST SD4 and NIST SD14 against a gallery of 1.1 million fingerprints. Empirical results demonstrate that DeepPrint is comparable to COTS matchers in accuracy at a significantly faster search speed.

- Benchmarking the authentication performance of DeepPrint on the NIST SD4 and NIST SD14 rolled-fingerprints databases and the FVC 2004 DB1 A slap fingerprint database [109]. Again, DeepPrint shows comparable performance against the two COTS matchers, demonstrating the generalization ability of DeepPrint to both rolled and slap fingerprint databases.

- Demonstrating that homomorphic encryption can be used to match DeepPrint templates in the encrypted domain, in real time (1.26 ms), with minimal loss to matching accuracy as shown for fixed-length face representations [14].

- An interpretability visualization which demonstrates our ability to guide DeepPrint to look at minutiae-related features.

4.2 Prior Work

Several early works [22, 89, 90] presented fixed-length fingerprint representations using traditional image processing techniques. In [89, 90], Jain *et al.* extracted a global fixed-length representation

Table 4.2 Published Studies on Fixed-Length Fingerprint Representations

Algorithm	HR @ PR = 1.0%[1] (NIST SD4)[2]	HR @ PR = 1.0% (NIST SD14)[3]	Template Size (bytes)	Gallery Size[4]
Fingercode [89, 90]	N.A.	N.A.	640	N.A.
MCC [22]	93.2%	91.0%	1,913	2,700
Inception v3 [18]	98.65%	98.93%	8,192	250,000
PDC [159]	93.3%	N.A.	N.A.	2,000
MDC [160]	99.2%	99.6%	1,200	2,700
Finger Patches [102]	**99.83%**	99.89%	1,024	2,700
DeepPrint (proposed)	99.75%	**99.93%**	**200**†	**1.1M**

[1] In some baselines we estimated the data points from a Figure (specific data points were not reported in the paper).
[2] Only 2,000 fingerprints are included in the gallery to enable comparison with previous works. (HR = Hit Rate, PR = Penetration Rate)
[3] Only last 2,700 pairs (2,700 probes; 2,700 gallery) are used to enable comparison with previous works.
[4] Largest gallery size used in the paper.
† The DeepPrint representation can be further compressed to only 64 bytes using product quantization with minor loss in accuracy.

of 640 bytes, called Fingercode, using a set of Gabor Filters. Cappelli *et al.* introduced a fixed-length minutiae descriptor, called Minutiae Cylinder Code (MCC), using 3D cylindrical structures computed with minutiae points [22]. While both of these representations demonstrated success at the time they were proposed, their accuracy is now significantly inferior to state-of-the-art COTS matchers

Following the seminal contributions of [89,90] and [22], the past 10 years of research on fixed-length fingerprint representations [16, 54, 96, 105, 122, 123, 164, 165, 184] has not produced a representation competitive in terms of fingerprint recognition accuracy with the traditional minutiae-based representation. However, recent studies [18, 102, 159, 160] have utilized deep networks to extract highly discriminative fixed-length fingerprint representations. More specifically, (i) Cao and Jain [18] used global alignment and Inception v3 to learn fixed-length fingerprint representations. (ii) Song and Feng [159] used deep networks to extract representations at various resolutions which were then aggregated into a global fixed-length representation. (iii) Song *et al.* [160] further learned fixed-length minutiae descriptors which were aggregated into a global fixed-length representation via an aggregation network. Finally, (v) Li *et al.* [102] extracted local descriptors from predefined “fingerprint classes” which were then aggregated into a global fixed-length representation through global average pooling.

While these efforts show tremendous promise, each method has some limitations. In particular, (i) the algorithms proposed in [18] and [159] both required computationally demanding global alignment as a preprocessing step, and the accuracy is inferior to state-of-the-art COTS matchers. (ii) The representations extracted in [160] require the arduous process of minutiae-detection, patch extraction, patch-level inference, and an aggregation network to build a single global feature representation. (iii) While the algorithm in [102] obtains high performance on rolled fingerprints (with small gallery size), the accuracy was not reported for slap fingerprints. Since [102] aggregates local descriptors by averaging them together, it is unlikely that the approach would work well when areas of the fingerprint are occluded or missing (often times the case in slap fingerprint databases like FVC 2004 DB1 A), and (v) all of the algorithms, suffer from lack of interpretability compared to traditional minutiae representations.

In addition, existing studies targeting deep, fixed-length fingerprint representations all lack an extensive, large-scale evaluation of the deep features. Indeed, one of the primary motivations for fixed-length fingerprint representations is to perform orders of magnitude faster large scale search. However, with the exception of Cao and Jain [18], who evaluate against a database of 250K fingerprints, the next largest gallery size used in any of the aforementioned studies is only 2,700.

As an addendum, deep networks have also been used to improve *specific sub-modules* of fingerprint recognition systems such as segmentation [32, 50, 124, 194], orientation field estimation [17, 140, 150], minutiae extraction [33, 125, 168], and minutiae descriptor extraction [19]. However, these works all still operate within the conventional paradigm of extracting an unordered, variable length set of minutiae for fingerprint matching.

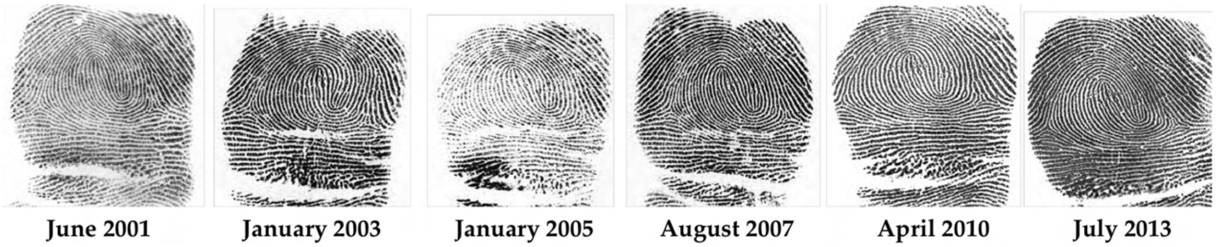


Figure 4.2 Fingerprint impressions from one subject in the DeepPrint training dataset [188]. Impressions were captured longitudinally, resulting in the variability across impressions (contrast and intensity from environmental conditions; distortion and alignment from user placement). Importantly, training with longitudinal data enables learning compact representations which are invariant to the typical noise observed across fingerprint impressions over time, a necessity in any fingerprint recognition system.

4.3 DeepPrint

In the following section, we (i) provide a high-level overview and intuition of DeepPrint, (ii) present how we incorporate automatic alignment into DeepPrint, and (iii) demonstrate how the accuracy and interpretability of DeepPrint is improved through the injection of fingerprint domain knowledge.

4.3.1 Overview

A high level overview of DeepPrint is provided in Figure 4.1 with pseudocode in Algorithm 2. DeepPrint is trained with a longitudinal database (Fig. 4.2) comprised of 455K rolled fingerprint images stemming from 38,291 unique fingers [188]. Longitudinal fingerprint databases consist of fingerprints from distinct subjects captured over time (Fig. 4.2) [188]. It is necessary to train DeepPrint with a large, longitudinal database so that it can learn compact, fixed-length representations which are invariant to the differences introduced during fingerprint image acquisition at different times and in different environments (humidity, temperature, user interaction with the reader, and finger injuries). The primary task during training is to predict the finger identity label $c \in [0, 38291]$ (encoded as a one-hot vector) of each of the 455K training fingerprint images ($\approx$ 12 fingerprint

Algorithm 2 Extract DeepPrint Representation

```
$L(I_f)$: Shallow localization network, outputs $x, y, \theta$
A: Affine matrix composed with parameters $x, y, \theta$
$G(I_f, A)$: Bilinear grid sampler, outputs aligned fingerprint
$S(I_t)$: Inception v4 stem
$E(\mathbf{x})$: Shared minutiae parameters
$M(\mathbf{x})$: Minutia representation branch
$D(\mathbf{x})$: Minutiae map estimation
$T(\mathbf{x})$: Texture representation branch

**Input:** Unaligned $448 \times 448$ fingerprint image $I_f$
$A \leftarrow (x, y, \theta) \leftarrow L(I_f)$
$I_t \leftarrow G(I_f, A)$
$F_{map} \leftarrow S(I_t)$
$M_{map} \leftarrow E(F_{map})$
$R_1 \leftarrow M(M_{map})$
$H \leftarrow D(M_{map})$
$R_2 \leftarrow T(F_{map})$
$\mathbf{R} \leftarrow R_1 \oplus R_2$
**Output:** Fingerprint representation $\mathbf{R} \in \mathbb{R}^{192}$ and minutiae-map $H$. ($H$ can be optionally
    utilized for (i) visualization and (ii) fusion of DeepPrint scores obtained via $\mathbf{R}$ with minutiae-
    matching scores.)
```

impressions / finger). The last fully connected layer is taken as the representation for fingerprint comparison during authentication and search.

The input to DeepPrint is a 448×448 [5] grayscale fingerprint image, I_f, which is first passed through the alignment module (Fig. 4.1). The alignment module consists of a localization network, L, and a grid sampler, G [83]. After applying the localization network and grid sampler to I_f, an aligned fingerprint I_t is passed to the base-network, S.

The base-network is the stem of the Inception v4 architecture (Inception v4 minus Inception modules). Following the base-network are two different branches (Fig. 4.1) comprised primarily of the three Inception modules (A, B, and C) described in [166]. The first branch, $T(x)$, completes

[5] Fingerprint images in our training dataset vary in size from $\approx\ 512 \times 512$ to $\approx\ 800 \times 800$. As a pre-processing step, we do a center cropping (using Gaussian filtering, dilation and erosion, and thresholding) to all images to $\approx\ 448 \times 448$. This size is sufficient to cover most of the rolled fingerprint area without extraneous background pixels.

the Inception v4 architecture [6] as $T(S(I_t))$ and performs the primary learning task of predicting a finger identity label directly from the cropped, aligned fingerprint I_t. It is included in order to learn the textural cues in the fingerprint image. The second branch (Figs. 4.1 and 4.5), $M(E(S(I_t)))$, again predicts the finger identity label from the aligned fingerprint I_t, but it also has a related side task (Fig. 4.5) of detecting the minutiae locations and orientations in I_t via $D(E(S(I_t)))$. In this manner, we guide this branch of the network to extract representations influenced by fingerprint minutiae (since parameters between the minutiae detection task and representation learning task are shared in $E(x)$). The textural cues act as complementary discriminative information to the minutiae-guided representation. The two 96-dimensional representations (each dimension is a float, consuming 4 bytes of space) are concatenated into a 192-dimensional representation (768 total bytes). Finally, the floats are truncated from 32 bits to 8 bit integer values, compressing the template size to 200 bytes (192 bytes for features and 8 bytes for 2 decompression parameters). Note that the minutiae set is not explicitly used in the final representation. Rather, we use the minutiae-map to guide our network training. However, for improved accuracy and interpretability, we can *optionally* store the minutiae set for use in a re-ranking scheme during large-scale search operations.

In the following subsections, we provide details of the major sub-components of the proposed network architecture.

4.3.2 Alignment

In nearly all fingerprint recognition systems, the first step is to perform alignment based on some reference points (such as the core point). However, this alignment is computationally expensive. This motivated us to adopt attention mechanisms such as the spatial transformers in [83].

The advantages of using the spatial transformer module in place of reference point based alignment algorithms are two-fold: (i) it requires only one forward pass through a shallow localization

[6] We selected Inception v4 after evaluating numerous other architectures such as: ResNet, Inception v3, Inception ResNet, and MobileNet.

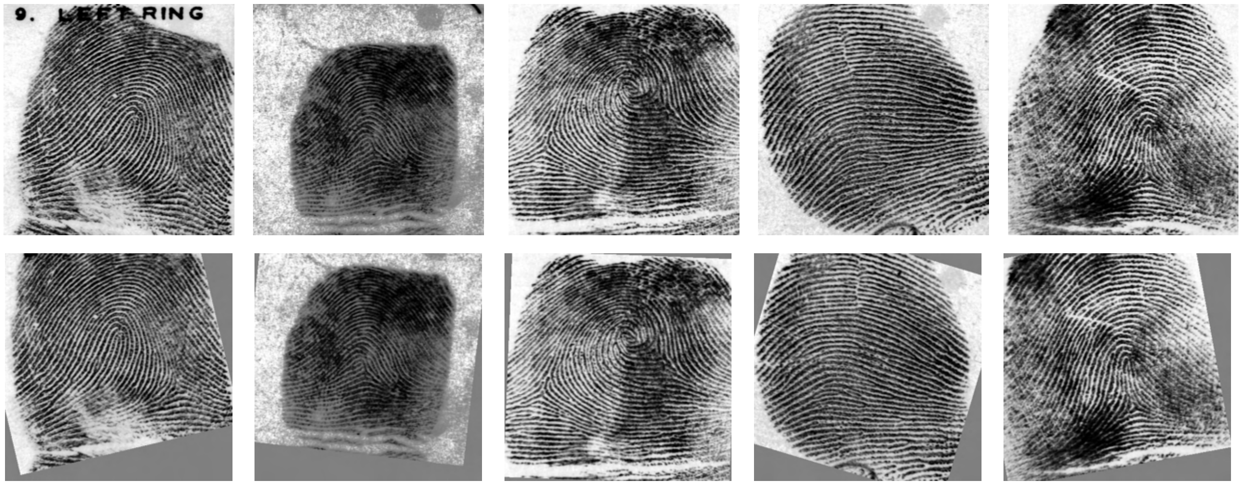


Figure 4.3 Unaligned fingerprint images from NIST SD4 (top row) and corresponding DeepPrint aligned fingerprint images (bottom row).

network (Table 4.3), followed by bilinear grid sampling. This reduces the computational complexity of alignment (we resize the 448×448 fingerprints to 128×128[7] to further speed up the localization estimation); (ii) The parameters of the localization network are tuned to minimize the loss (Eq. 4.3.9) of the base-network and representation extraction networks. In other words, rather than supervising the transformation via reference points (such as the core point), we let the base-network and representation extraction networks tell the localization network what a "good" transformation is, so that it can learn a more discriminative representation for the input fingerprint.

Given an unaligned fingerprint image I_f, a shallow localization network first hypothesizes the translation and rotation parameters (x,y, and θ) of an affine transformation matrix A_θ (Fig. 4.1). A user specified scaling parameter λ is used to complete A_θ (Fig. 4.1). This scaling parameter stipulates the area of the input fingerprint image which will be cropped. We train two DeepPrint models, one for rolled fingerprints ($\lambda = 1$) and one for slap fingerprints ($\lambda = \frac{285}{448}$) meaning a 285×285 fingerprint area window will be cropped from the 448×448 input fingerprint image. Given A_θ, a grid sampler G samples the input image I_f pixels (x_i^f, y_i^f) for every target grid location (x_i^t, y_i^t) to output the aligned fingerprint image I_t in accordance with Equation 4.3.1.

[7] We also tried 64×64, however, we could not obtain consistent alignment at this resolution.

Table 4.3 Localization Network Architecture

Type	Output Size	Filter Size, Stride
Convolution	$128 \times 128 \times 24$	5×5, 1
Max Pooling	$64 \times 64 \times 24$	2×2, 2
Convolution	$64 \times 64 \times 32$	3×3, 1
Max Pooling	$32 \times 32 \times 32$	2×2, 2
Convolution	$32 \times 32 \times 48$	3×3, 1
Max Pooling	$16 \times 16 \times 48$	2×2, 2
Convolution	$16 \times 16 \times 64$	3×3, 1
Max Pooling	$8 \times 8 \times 64$	2×2, 2
Fully Connected	64	
Fully Connected	3†	

† These three outputs correspond to x,y,θ shown in Fig. 4.1.

$$\begin{pmatrix} x_i^f \\ y_i^f \\ 1 \end{pmatrix} = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{4.3.1}$$

Once I_t has been computed, it is passed on to the base-network for classification. Finally, the parameters for the localization network are updated based upon the loss in Equation 4.3.9.

The architecture used for our localization network is shown in Table 4.3 and images from before and after the alignment module are shown in Figure 4.3. In order to get the localization network to properly converge, (i) the learning rate was scaled by 0.035 and (ii) the upper bound of the estimated affine matrix translation and rotation parameters was set to 224 pixels and ± 60 degrees, respectively. These constraints are based on our domain knowledge on the maximum extent a user would rotate or translate their fingers during placement on the reader platen.

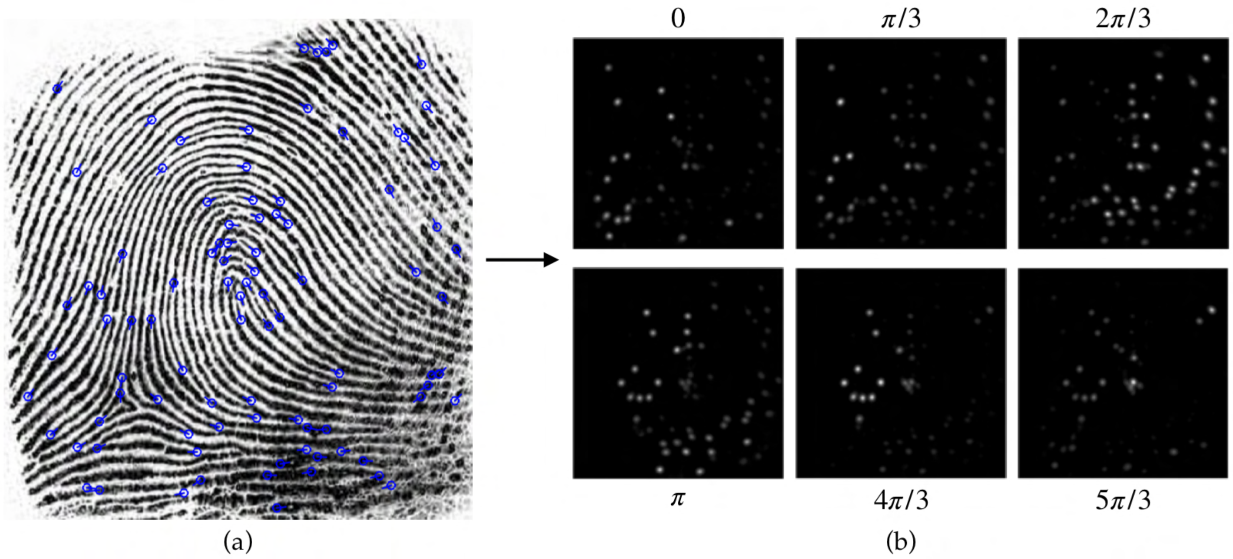


Figure 4.4 Minutiae Map Extraction. The minutiae locations and orientations of an input fingerprint (a) are encoded as a 6-channel minutiae map (b). The "hot spots" in each channel indicate the spatial location of the minutiae points. The k^{th} channel of the hot spots indicate the contributions of each minutiae to the $k\pi/3$ orientation.

4.3.3 Minutiae Map Domain Knowledge

To prevent overfitting the network to the training data and to extract interpretable deep features, we incorporate fingerprint domain knowledge into DeepPrint. The specific domain knowledge we incorporate into our network architecture is hereafter referred to as the *minutiae map* [20]. Note that the minutiae map is not explicitly used in the fixed-length fingerprint representation, but the information contained in the map is indirectly embedded in the network during training.

A minutiae map is essentially a 6-channel heatmap quantizing the locations (x, y) and orientations $\theta \in [0, 2\pi]$ of the minutiae within a fingerprint image. More formally, let h and w be the height and width of an input fingerprint image and $T = \{m_1, m_2, ..., m_n\}$ be its minutiae template with n minutiae points, where $m_t = (x_t, y_t, \theta_t)$ and $t = 1, ..., n$. Then, the minutiae map $H \in \mathbb{R}^{h \times w \times 6}$ at (i, j, k) can be computed by summing the location and orientation contributions of each of the minutiae in T to obtain the heat map (Fig. 4.4 (b)).

$$H(i,j,k) = \sum_{t=1}^{n} C_s((x_t, y_t),(i,j)) \cdot C_o(\theta_t, 2k\pi/6) \tag{4.3.2}$$

where $C_s(.)$ and $C_o(.)$ calculate the spatial and orientation contribution of minutiae m_t to the minutiae map at (i,j,k) based upon the euclidean distance of (x_t, y_t) to (i,j) and the orientation difference between θ_t and $2k\pi/6$ as follows:

$$C_s((x_t, y_t),(i,j)) = exp(-\frac{||(x_t, y_t) - (i,j)||_2^2}{2\sigma_s^2}) \tag{4.3.3}$$

$$C_o(\theta_t, 2k\pi/6) = exp(-\frac{d\phi(\theta_t, 2k\pi/6)}{2\sigma_s^2}) \tag{4.3.4}$$

where σ_s^2 is the parameter which controls the width of the gaussian, and $d\phi(\theta_1, \theta_2)$ is the orientation difference between angles θ_1 and θ_2:

$$d\phi(\theta_1, \theta_2) = \begin{cases} |\theta_1 - \theta_2| & -\pi \leq \theta_1 - \theta_2 \leq \pi \\ 2\pi - |\theta_1 - \theta_2| & otherwise. \end{cases} \tag{4.3.5}$$

An example fingerprint image and its corresponding minutiae map are shown in Figure 4.4. A minutiae-map can be converted back to a minutiae set by finding the local maximums in a channel (location), and individual channel contributions (orientation), followed by non-maximal suppression to remove spurious minutiae[8].

4.3.4 Multi-Task Architecture

The minutiae-map domain knowledge is injected into DeepPrint via multitask learning. Multitask learning improves generalizability of a model since domain knowledge within the training signals of related tasks acts as an inductive bias [23,186]. The multi-task branch of the DeepPrint architecture is shown in Figures 4.1 and 4.5. The primary task of the branch is to extract a representation

[8] Code for converting a minutiae set to a minutiae map and vice versa is open-sourced: https://bit.ly/2KpbPxV

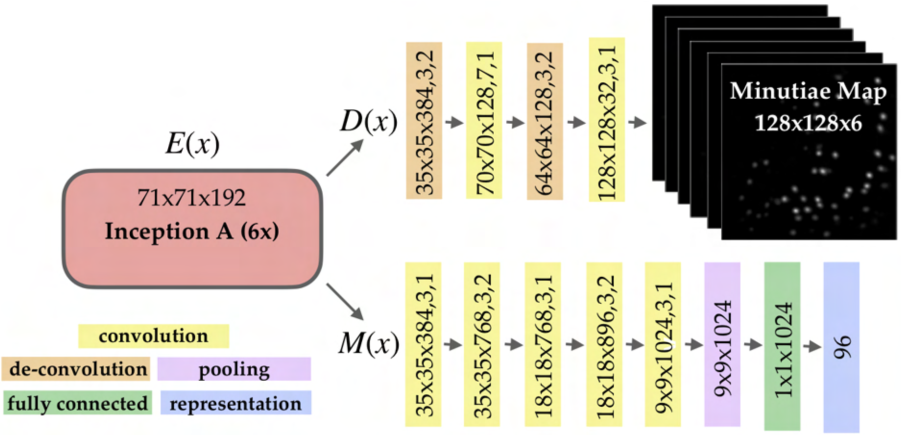


Figure 4.5 The custom multi-task **minutiae branch** of DeepPrint. The dimensions inside each box represent the input dimensions, kernel size, and stride length, respectively.

and subsequently classify a given fingerprint image into its “finger identity”. The secondary task is to estimate the minutiae-map. Since parameters are shared between the representation learning task and the minutiae-map extraction task, we guide the minutiae-branch of our network to extract fingerprint representations that are influenced by minutiae locations and orientations. At the same time, a separate branch in DeepPrint aims to extract a complementary texture-based representation by directly predicting the identity of an input fingerprint without any domain knowledge (Fig. 4.1). DeepPrint extracts minutiae maps of size $128 \times 128 \times 6$ [9] to encode the minutiae locations and orientations of an input fingerprint image of size $448 \times 448 \times 1$. The ground truth minutiae maps for training DeepPrint are estimated using the open source minutiae extractor proposed in [20].

Note, we combine the texture branch with the minutiae branch in the DeepPrint architecture (rather than two separate networks) for the following reasons: (i) the minutiae branch and the texture branch share a number of parameters (the Inception v4 stem), reducing the model complexity that two separate models would necessitate, and (ii) the spatial transformer (alignment module) is optimized based on both branches (*i.e.* learned alignment benefits both the texture-based and minutiae-based representations) avoiding two separate spatial transformer modules and alignments.

[9] We extract maps of $128 \times 128 \times 6$ to save GPU memory during training (enabling a larger batch size), and to reduce disk space requirements for storage of the maps.

More formally, we incorporate domain knowledge into the DeepPrint representation by computing the network's loss in the following manner. First, given R_1 and R_2 as computed in Algorithm 2, fully connected layers are applied for identity classification logits, outputting $\mathbf{y}_1 \in \mathbb{R}^c$ and $\mathbf{y}_2 \in \mathbb{R}^c$, where c is the number of identities in the training set. Next, $\mathbf{y}_1$ and $\mathbf{y}_2$ are both passed to a softmax layer to compute the probabilities $\hat{\mathbf{y}}_1$ and $\hat{\mathbf{y}}_2$ of R_1 and R_2 belonging to each identity. Finally, $\hat{\mathbf{y}}_1$ and $\hat{\mathbf{y}}_2$, the ground truth label y, and the network's parameters w, can be used to compute the combined cross-entropy loss of the two branches and an image I_t:

$$\mathcal{L}_1(I_t, y) = -log(\hat{\mathbf{y}}_1^{j=y}|I_t, w) - log(\hat{\mathbf{y}}_2^{j=y}|I_t, w) \tag{4.3.6}$$

where $j \in \{1, \cdots, c\}$. To further reduce the intra-class variance of the learned features, we also employ the widely used center-loss first proposed in [179] for face recognition. In particular, we compute two center-loss terms, one for each branch in our multi-task architecture as:

$$\mathcal{L}_2(I_t) = ||R_1 - ctr_1^n||_2^2 + ||R_2 - ctr_2^n||_2^2 \tag{4.3.7}$$

where ctr_i^n, are the branch, i, and subject, n, specific centers for a fingerprint image I_t.

For computing the loss of the minutiae map estimation side task, we employ the Mean Squared Error Loss between the estimated minutiae map $\mathbf{H}$ and the ground truth minutiae map [10] H as follows:

$$\mathcal{L}_3(I_t, H) = \sum_{i,j,k} (\mathbf{H}_{i,j,k} - H_{i,j,k})^2 \tag{4.3.8}$$

Finally, using the addition of all these loss terms, and a dataset comprised of N training images, our model parameters w are trained in accordance with:

$$\arg\min_{w} \sum_{i=1}^{N} \lambda_1 \mathcal{L}_1(I_t^i, y^i) + \lambda_2 \mathcal{L}_2(I_t^i) + \lambda_3 \mathcal{L}_3(I_t^i, H^i) \tag{4.3.9}$$

[10] The ground truth minutiae maps are estimated using the open-source minutiae extractor in [20].

where $\{\lambda_1 = 1, \lambda_2 = 0.00125, \lambda_3 = 0.095\}$ are empirically set to obtain convergence. Note, during the training, we augment our dataset with random rotations, translations, brightness, and cropping. We use the RMSProp optimizer with a batch size of 30. Weights are initialized with the variance scaling initializer. Regularization included dropout (before the embedding fully connected layer) with a keep probability of 0.8 and weight decay of 0.00004. We trained for 140K steps, which lasted 25 hours.

After the multitask architecture has converged, a fixed length feature representation can be acquired by extracting the fully connected layer before the softmax layers in both of the network's branches. Let $R_1 \in \mathbb{R}^{96}$ be the unit-length minutiae representation and $R_2 \in \mathbb{R}^{96}$ be the unit-length texture representation. Then, a final feature representation is obtained by concatenation of R_1 and R_2 into $\mathbf{R} \in \mathbb{R}^{192}$, followed by normalization of $\mathbf{R}$ to unit length.

4.3.5 Template Compression

The final step in the DeepPrint representation extraction is template compression. In particular, the 192-dimensional DeepPrint representation consumes a total of 768 bytes. We can compress this size to 200 bytes by truncating the 32 bit floating point feature values to 8-bit integer values in the range of [0,255] using min-max normalization. In particular, given a DeepPrint representation $\mathbf{R} \in \mathbb{R}^{192}$, we transfer the domain of $\mathbf{R}$ to $\mathbf{R} \in \mathbb{N}^{192}$ and output $\mathbf{R}'$, where we restrict the set of the natural numbers $\mathbb{N}$ to the range of [0,255]. More formally:

$$\mathbf{R}' = \left\lfloor \frac{255(\mathbf{R} - min(\mathbf{R}))}{max(\mathbf{R}) - min(\mathbf{R})} \right\rfloor \tag{4.3.10}$$

where $min(\mathbf{R})$ and $max(\mathbf{R})$ output the minimum and maximum feature values of the vector $\mathbf{R}$, respectively. In order to decompress the features back to float values for matching, we need to save the minimum and maximum values for each representation. Thus, our final representation is 200 bytes, 192 bytes for the features, 4 bytes for the minimum value and 4 bytes for the maximum value. To decompress the representations (when loading them into RAM), we simply reverse the min-max normalization using the saved minimum and maximum values. Table **??** shows that compression only minimally impacts the matching accuracy.

Table 4.4 Effect of Compression on Accuracy

Dataset	DeepPrint Uncompressed Features	DeepPrint Compressed Features
NIST SD4[†]	97.95%	97.90%
FVC 2004 DB1 A[†]	97.53%	97.50%

[†] TAR @ FAR = 0.01% is reported.
[†] TAR @ FAR = 0.1% is reported.

4.4 DeepPrint Matching

Two, unit length, DeepPrint representations $\mathbf{R}_p$ and $\mathbf{R}_g$ can be easily matched using the cosine similarity between the two representations. In particular:

$$s(\mathbf{R}_p, \mathbf{R}_g) = \mathbf{R}_p^{\intercal} \cdot \mathbf{R}_g \tag{4.4.1}$$

Thus, DeepPrint authentication (1:1 matching) requires only 192 multiplications and 191 additions. We also experimented with euclidian distance as a scoring function, but consistently obtained higher performance with cosine similarity. Note that if compression is added, there would be an additional d subtractions and d multiplications to reverse the min-max normalization of the enrolled representation. Therefore, the authentication time effectively doubles. However, depending on the application or implementation, compression does not necessarily effect the search speed since the gallery of representations could be already decompressed and in RAM before performing a search.

4.4.1 Fusion of DeepPrint Score with Minutiae Score

Given the speed of matching two DeepPrint representations, the minutiae-based match scores of any existing AFIS can also be fused together with the DeepPrint scores with minimal loss to the overall AFIS authentication speed (*i.e.* DeepPrint can be easily used as an add-on to existing minutiae-based AFIS to improve recognition accuracy). In our experimental analysis, we demonstrate this by fusing DeepPrint scores together with the scores of minutiae-based matchers COTS A, COTS B, and [20] and subsequently improving authentication accuracy. This indicates that the information contained in the compact DeepPrint representation is complementary to that of minutiae representations. Note, since DeepPrint already extracts minutiae as a side task, fusion with a minutiae-based matcher requires little extra computational overhead (simply feed the minutiae extracted by DeepPrint directly to the minutiae matcher, eliminating the need to extract minutiae a second time).

4.5 DeepPrint Search

Fingerprint search entails finding the top k candidates, in a database (gallery or background) of N fingerprints, for an input probe fingerprint. The simplest algorithm for obtaining the top k candidates is to (i) compute a similarity measure between the probe template and every enrolled template in the database, (ii) sort the enrolled templates by their similarity to the probe [11], and (iii) select the top k most similar enrollees. More formally, finding the top k candidates $C_k(.)$ in a gallery G for a probe fingerprint $\mathbf{R}_p$ is formulated as:

$$C_k(\mathbf{R}_p) = Rank_k(\{s(\mathbf{R}_p, \mathbf{R}_g)|\mathbf{R}_g \in G\}) \tag{4.5.1}$$

where $Rank_k(.)$ returns the k most similar candidates from an input set of candidates and s is a similarity function such as defined in Equation 4.4.1.

[11] In our search experiments, we reduce the typical sorting time from $Nlog(N)$ to $Nlog(k)$ (where $k << N$) by maintaining a priority queue of size k since we only care about the scores of the top k candidates. This trick reduces sorting time from 23 seconds to 8 seconds when the gallery size $N = 100,000,000$ and the candidate list size $k = 100$.

Since minutiae-based matching is computationally expensive, comparing the probe to every template enrolled in the database in a timely manner is not feasible with minutiae matchers. This has led to a number of schemes to either significantly reduce the search space, or utilize high-level features to quickly index top candidates [12, 21, 95, 106, 163]. However, such methods have not achieved high-levels of accuracy on public benchmark datasets such as NIST SD4 or NIST SD14.

In contrast to minutiae-matchers, the fixed-length, 200 byte DeepPrint representations can be matched extremely quickly using Equation 4.4.1. Therefore, large scale search with DeepPrint can be performed by *exhaustive* comparison of the probe template to every gallery template in accordance with Equation 4.5.1. The complexity of exhaustive search is linear with respect to both the gallery size N and the dimensionality d of the DeepPrint representation ($d = 192$ in this case).

4.5.1 Faster Search

Although exhaustive search can be effectively utilized with DeepPrint representations in conjunction with Equation 4.5.1, it may be desirable to even further decrease the search time. For example, when searching against 100 million fingerprints, the DeepPrint search time is still (11 seconds on an i9 processor with 64 GB of RAM) [12]. A natural way to reduce the search time further with minimal loss to accuracy is to utilize an effective approximate nearest neighbor (ANN) algorithm.

Product Quantization is one such ANN algorithm which has been successfully utilized in large-scale face search [177]. Product quantization is still an exhaustive search algorithm, however, representations are first compressed via keys to a lookup table, which significantly reduces the comparison time between two representations. In other words, product quantization reformulates the comparison function in Equation 4.4.1 to a series of lookup operations in a table stored in RAM. More formally, a given DeepPrint representation $\mathbf{R}_g$ of dimensionality d, is first decomposed into m sub-vectors as:

[12] Search time for 100 million gallery was simulated by generating 100 million random representations, where each feature was a 32-bit float value drawn from a uniform distribution from 0 to 1.

$$\mathbf{R}_g = (R^1, R^2, ..., R^m) \tag{4.5.2}$$

Next, each m^{th} sub-vector $R^i \in \mathbb{R}^{d/m}$ is mapped to a codeword c^i_j in a codebook $\mathcal{C}^i = \{c^i_{j=1,2,...,z} | c^i_j \in \mathbb{R}^{d/m}\}$ where z is the size of the codebook. The index j of each codeword c^i_j can be represented as a binary code of $log_2(z)$ bits. Therefore, after mapping each sub-vector to its codeword, the original d-dimensional representation $\mathbf{R}_g$ ($d = 192$ for DeepPrint) can be compressed to only $m * log_2(z)$ bits!

The codewords $c^i_j \in \mathbb{R}^{d/m}$ for each codebook $\mathcal{C}^i$ are computed offline (before search time) using k-means clustering for each sub-vector. Thus each codebook $\mathcal{C}^i$ contains z centroids computed from the corresponding sub-vectors R^i. Given all m codebooks $\{\mathcal{C}^1, \mathcal{C}^2, \mathcal{C}^3, ..., \mathcal{C}^m\}$, the product quantizer of $\mathbf{R}_g$ is computed as:

$$q(\mathbf{R}_g) = (q^1(R^1), ..., q^m(R^m)) \tag{4.5.3}$$

where $q^i(R^i)$ is the index of the nearest centroid in the codebook $\mathcal{C}^i, i = 1, ..., m$.

Finally, given a DeepPrint probe representation $\mathbf{R}_p$, and the now quantized gallery template $\mathbf{R}_g$, a match score can be obtained in accordance with Equation 4.5.4:

$$s(\mathbf{R}_p, \mathbf{R}_g) = ||\mathbf{R}_p - q(\mathbf{R}_g)||_2^2 = \sum_{i=1}^{m} ||\mathbf{R}^i_p - q^i(R^i)||_2^2 \tag{4.5.4}$$

Thus matching a probe template to each quantized template in the gallery requires a one-time build up of a $m \times z$ table which is stored in RAM, followed by m lookups and additions for each quantized template in the gallery. In our experiments, we set $z = 256$ and $m = 64$. A quantized template in the gallery is compressed to 64 bytes, and search is reduced from 192 additions and multiplications (N times, where N is the gallery size) to a one-time $m \times z$ table build up, followed by 64 lookups and additions for each gallery template (a significant savings on memory and search time)[13].

[13] We used the Facebook Faiss PQ implementation: https://github.com/facebookresearch/faiss

4.5.2 Two-stage DeepPrint Search

In addition to increasing the speed of large-scale fingerprint search using DeepPrint with product quantization, we also propose a method whereby a negligible amount of search speed can be sacrificed in order to further improve the search accuracy. In particular, we first use the DeepPrint representations to find the top-k[14] candidates for a probe $\mathbf{R}_p$ in a gallery G. Then, the top-k candidates are re-ranked using the scores of a minutiae-matcher fused together with the DeepPrint similarity scores. More formally, given a minutiae-matcher function $m(.)$, the k re-ranked candidates can be computed by:

$$Sort_k(\{m(m_p, m_g) + s(\mathbf{R}_p, \mathbf{R}_g)|g \in G\}) \tag{4.5.5}$$

where m_p and m_g two varying length minutiae templates, $\mathbf{R}_p$ and $\mathbf{R}_g$ are the two fixed-length DeepPrint templates, $s(.)$ is the DeepPrint similarity score (either Equation 4.4.1 or Equation 4.5.4), and $Sort_k$ returns a list of k candidates sorted in descending order by similarity score.

We note that since DeepPrint already outputs a minutiae-map, which can easily be converted to a minutiae-set, fusing DeepPrint with a minutiae matcher is quite seamless. We simply convert the DeepPrint minutiae-maps to minutiae-sets, and subsequently input the minutiae-sets to a minutiae-matcher such as the open-source minutiae matcher in [20].

4.6 Secure DeepPrint Matching

One of the primary benefits of the fixed-length, 192-dimensional DeepPrint representation is that it can be encrypted and matched in the encrypted domain (with 192 bits of security [14]) with fully homomorphic encryption (FHE). In particular, FHE enables performing any number of both addition and multiplication operations in the encrypted domain. Since DeepPrint representations can be matched using only multiplication and addition operations (Eq. 4.4.1), they can be matched

[14] The value of k depends on the gallery size N. For the gallery size of $N = 1.1$ million, we empirically selected $k = 500$.

in the encrypted domain with minimal loss to system accuracy (only loss in accuracy comes from converting floating point features to integer value features, resulting in a loss of precision).

In contrast, minutiae-based representations cannot be matched under FHE, since the matching function cannot be reduced to simple addition and multiplication operations. Furthermore, existing encryption schemes for minutiae-based templates such as the fuzzy-vault, result in a loss of matching accuracy, and are very sensitive to fingerprint pre-alignment [172]. We demonstrate in our experiments that the DeepPrint authentication performance remains almost unaltered following FHE matching. We utilize the Fan-Vercauteren FHE Scheme [53] with improvements from [14] for improved speed and efficiency[15].

4.7 Datasets

We use four sources of data in our experiments. Our training data is a longitudinal dataset comprised of 455K rolled fingerprint images from 38,291 unique fingers taken from [188]. Our testing data is comprised of both large area rolled fingerprint images taken from NIST SD4 and NIST SD14 (similar to the training data) and small area slap fingerprint images from FVC 2004 DB1 A.

4.7.1 NIST SD4 & NIST SD14

The NIST SD4 and NIST SD14 databases are both comprised of rolled fingerprint images (Fig. 4.6). Due to the number of challenging fingerprint images contained in both datasets (even for commercial matchers), they continue to be popular benchmark datasets for automated fingerprint recognition algorithms. NIST SD4 is comprised of 2,000 unique fingerprint pairs (total of 4,000 images), evenly distributed across the 5 fingerprint types (arch, left loop, right loop, tented arch, and whorl). NIST SD14 is a much larger dataset comprised of 27,000 unique fingerprint pairs. However, in most papers published on fingerprint search, only the last 2,700 pairs from

[15] We use the following open-source implementation: https://github.com/human-analysis/secure-face-matching

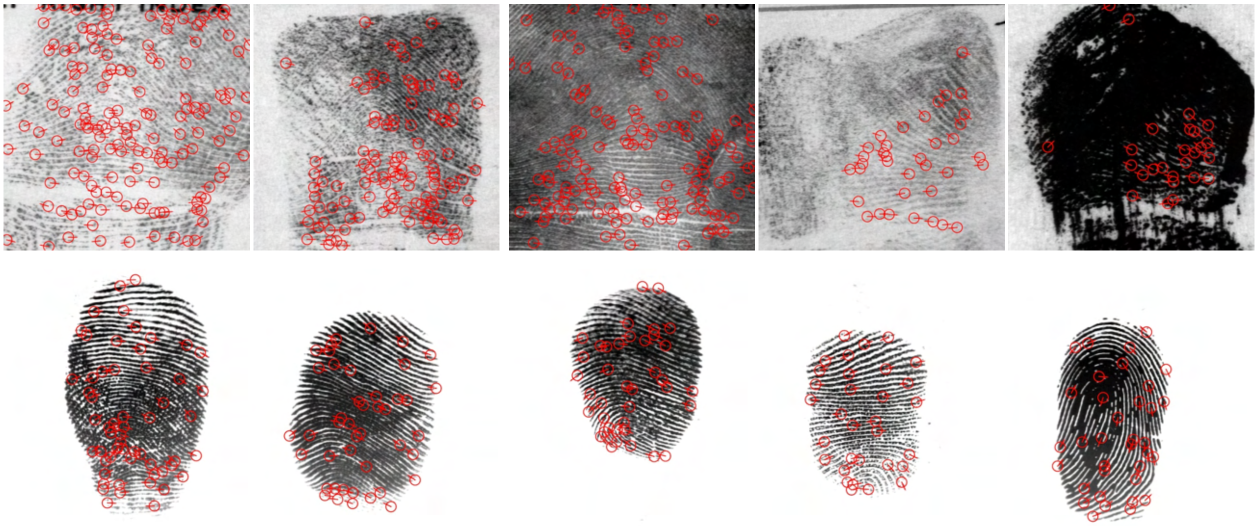


Figure 4.6 Examples of poor quality fingerprint images from benchmark datasets. Row 1: Rolled fingerprint impressions from NIST SD4. Row 2: Slap fingerprint images from FVC 2004 DB1 A. Rolled fingerprints are often heavily smudged, making them challenging to accurately recognize. FVC 2004 DB1 A also has several distinct challenges such as small overlapping fingerprint area between two fingerprint images, heavy non-linear distortions, and extreme finger conditions (wet or dry). Minutiae annotated with COTS A.

NIST SD14 are utilized for evaluation. To fairly compare DeepPrint with previous approaches, we also use the last 2,700 pairs of NIST SD14 for evaluation.

4.7.2 FVC 2004 DB1 A

The FVC 2004 DB1 A dataset is an extremely challenging benchmark dataset (even for commercial matchers) for several reasons: (i) small overlapping fingerprint area between fingerprint images from the same subject, (ii) heavy non-linear distortion, and (iii) extremely wet and dry fingers (Fig. 4.6). Another major motivation for selecting FVC 2004 DB1 A as a benchmark dataset is that it is comprised of slap fingerprint images. Because of this, we are able to demonstrate that even though DeepPrint was trained on rolled fingerprint images similar to NIST SD4 and NIST SD14, our incorporation of domain knowledge into the network architecture enables it to generalize well to slap fingerprint datasets.

4.8 COTS Matchers

In most all of our experiments, we benchmark DeepPrint against COTS A and COTS B (Verifinger 10.0 or Innovatrics v7.2.1.40, the latest version of the SDK as of July, 2019). Due to our Non-disclosure agreement, we cannot provide a link between aliases COTS A and COTS B and Verifinger or Innovatrics. Both of these SDKs provide an ISO minutia-only template as well as a proprietary template comprised of minutiae and other features. To obtain the best performance from each SDK, we extracted the more discriminative proprietary templates. The proprietary templates are comprised of minutiae and other features unknown to us. We note that both Verifinger and Innovatrics are top performers in the NIST and FVC evaluations [107, 178].

4.9 Benchmark Evaluations

We begin our experiments by comparing the DeepPrint search performance to the state-of-the-art fixed-length representations reported in the literature. Then, we show that the DeepPrint representation can also be used for state-of-the-art authentication by benchmarking against two of the top COTS fingerprint matchers in the market. We further show that this authentication can be performed in the encrypted domain using fully homomorphic encryption. Finally, we conclude our experiments by benchmarking the large-scale search accuracy of the DeepPrint representation against the same two COTS search algorithms.

4.9.1 Search (1:N Comparison)

Our first experimental objective is to demonstrate that the fixed-length DeepPrint representation can compete with the best fixed-length representations reported in the academic literature [18, 102] in terms of its search accuracy on popular benchmark datasets and protocols. In particular, we compute the Rank-1 search accuracy of the DeepPrint representation on both NIST SD4 and the last 2,700 pairs of NIST SD14 to follow the protocol of the earlier studies.

Table 4.5 Benchmarking DeepPrint Search Accuracy against Fixed-Length Representations in the Literature and COTS

Algorithm[†]	Template Description	NIST SD4[1] Rank-1 (%)	NIST SD14[2] Rank-1 (%)	Template Size Range (kB)	Search Time (ms)[3]
Inception v3 + COTS [18]	Fixed-Length	97.80	N.A.	8	175
Finger Patches [102]	Fixed-Length	99.27	99.04	1.0	16
DeepPrint	Fixed-Length	98.70	99.22	**0.2**	**11**
COTS A	Minutiae-based[4]	**99.55**	**99.92**	(1.5,23.7)	72
COTS B	Minutiae-based[4]	92.9	92.6	(0.6,5.3)	20

[1] Only 2,000 fingerprints are included in the gallery to enable comparison with previous works.
[2] Last 2,700 pairs are used to enable comparison with previous works.
[3] Search times for all algorithms benchmarked on NIST SD4 with an Intel Core i9-7900X CPU @ 3.30GHz
[4] We use the proprietary COTS templates which are comprised of minutiae together with other proprietary features.
[†] These results primarily show that (i) DeepPrints is competitive with the best fixed-length representation in the literature [102] (with a smaller template size) and state-of-the-art COTS, but also (ii) the benchmark dataset performances are saturated due to small gallery sizes. Therefore, in subsequent experiments we compare with state-of-the-art COTS against a background of 1.1 million.

The results, reported in Table 4.5, indicate that the DeepPrint representation is competitive with the most accurate search algorithm previously published in [102] (slightly lower performance on NIST4 and slightly higher on NIST14). However, we also note that the existing benchmarks (NIST SD4 and NISTSD14) for fingerprint search have now become saturated, making it difficult to showcase the differences between published approaches. Therefore, in subsequent experiments, we better demonstrate the efficacy of the DeepPrint representation by evaluating against a background of 1.1 million fingerprints (instead of the $\approx 2K$ in existing benchmarks).

We highlight once again that DeepPrint has the smallest template among state-of-the-art fixed length representations (200 bytes vs 1,024 bytes for the next smallest).

The search performance on FVC 2004 DB1 A is not reported, since the background is not of sufficient size (only 700 slap prints) to provide any meaningful search results.

4.9.2 Authentication

We benchmark the authentication performance of DeepPrint against two state-of-the-art COTS minutiae-based matchers, namely COTS A and COTS B. We note that none of the more recent works on fixed-length fingerprint representation [18, 102, 159, 160] have considered authentication performance, making it difficult for us to compare with these approaches (to the best of our knowledge, the code for these methods is not open-sourced).

From the experimental results (Tables 4.6 and 4.7), we note that DeepPrint outperforms COTS B on all benchmark testing protocols. We further note that DeepPrint outperforms both COTS A and COTS B on the very challenging FVC 2004 DB1 A (Fig. 4.6). The ability of DeepPrint to surpass COTS A and COTS B on the FVC slap fingerprint dataset is a very exciting find, given the DeepPrint network was trained on rolled fingerprint images which are comprised of very different textural characteristics than slap fingerprint impressions (Fig. 4.6). In comparison to rolled fingerprints, slap fingerprints often (i) require more severe alignment, (ii) can contain heavier non-linear distortion, (iii) and are much smaller with respect to impression area. We posit that our injection of domain knowledge (both alignment and minutiae detection) into the DeepPrint architecture help it to generalize well from the rolled fingerprints it was trained on to the slap fingerprints comprising FVC 2004 DB1 A. We demonstrate this further in a later ablation study.

Table 4.6 Authentication Accuracy (FVC 2004 DB1 A)

DeepPrint	COTS A	COTS B	DeepPrint + COTS A[1]	DeepPrint + COTS B[1]	DeepPrint + [20][1]
97.5%[†]	96.75%	96.57%	**98.93%**	98.46%	97.6%

[1] Sum score fusion is used.

[†] TAR @ FAR of 0.1% is reported since there are only 4,950 imposter pairs in the FVC protocol.

Table 4.7 Authentication Accuracy (Rolled-Fingerprints)

Algorithm	NIST SD4 TAR @ FAR = 0.01%	NIST SD14 TAR @ FAR = 0.01%
COTS A	**99.70**	**99.89**
COTS B	97.80	97.85
Cao *et al.* [20][1]	96.75	95.96
DeepPrint	97.90	98.55
DeepPrint + Minutiae [20]	98.70	99.0

[1] Minutiae extracted from DeepPrint minutiae-map ($\mathbf{H}$) and fed directly into minutiae matcher proposed in [20].

4.9.2.1 Fusion with Minutiae-Matchers

Another interesting result with respect to the DeepPrint authentication performance is that of the score distributions. In particular, we found that minutiae-based matchers COTS A and COTS B have very peaked imposter distributions near 0. Indeed, this is very typical of minutiae-matchers. In contrast, DeepPrint, has a peaked genuine distribution around 1.0, and a much flatter imposter distribution. In other words, COTS is generally stronger at true rejects, while DeepPrint is stronger at true accepts. This complementary phenomena motivated us to fuse DeepPrint with minutiae-based matchers to further improve their authentication performance (Table 4.6). Indeed, our results (Table 4.6) indicate that the DeepPrint representation does contain features complementary to minutiae-based matchers, given the improvement in authentication performance under score level fusion. We note that since a DeepPrints score can be computed with only 192 multiplications and 191 additions, it requires very little overhead for existing COTS matchers to integrate the DeepPrint representation into their matcher.

4.9.2.2 Secure Authentication

In addition to being competitive in authentication accuracy with state-of-the-art minutiae matchers, the *fixed-length* DeepPrint representation also offers the distinct advantage of matching in the encrypted domain (using FHE). Here we verify that the DeepPrint authentication accuracy remains intact following encryption. We also benchmark the authentication speed in the encrypted domain. Our empirical results (Table 4.8) demonstrate that the authentication accuracy remains nearly the same following FHE, and that authentication between a pair of templates takes only **1.26** milliseconds in the encrypted domain.

Table 4.8 Encrypted Authentication using DeepPrint Representation

Algorithm	NIST SD4[2]	NIST SD14[2]	FVC 2004 DB1 A[3]
DeepPrint	97.9%	98.55%	97.5%
DeepPrint + FHE[1]	96.9%	97.3%	97.0%

[1] Fully homomorphic encryption is utilized (match time: 1.26 ms).
[2] TAR @ FAR = 0.01%. [3] TAR @ FAR = 0.1%

4.10 Large Scale Search

Perhaps the most important attribute of the compact DeepPrint representation is its ability to perform extremely fast fingerprint search against large galleries. To adequately showcase this feature, we benchmark the DeepPrint search accuracy against COTS A and COTS B on a gallery of over 1.1 million rolled fingerprint images. The experimental results show that DeepPrint is able to obtain competitive search accuracy with the top COTS algorithm, at orders of magnitude faster speeds. Note, we are unable to benchmark other recent fixed-length representations in the literature against the large scale background, since code for these algorithms has not been open-sourced.

4.10.1 DeepPrint Search

First, we show the search performance of DeepPrints using a simple exhaustive search technique previously described. In particular, we match a probe template to every template in the gallery, and select the k candidates with the highest similarity scores. We use the NIST SD4 and NIST SD14 databases in conjunction with a gallery of 1.1 million rolled fingerprints. Under this exhaustive search scheme, the DeepPrint representation enables obtaining Rank-1 identification accuracies of **95.15% and 94.44%**, respectively (Table 4.10) and (Fig. 4.7). Notably, the search time is only **160 milliseconds**. At Rank-100, the search accuracies for both datasets cross over 99%. In our subsequent experiments, we demonstrate how we can re-rank the top k candidates to further improve the Rank-1 accuracy with minimal cost to the search time.

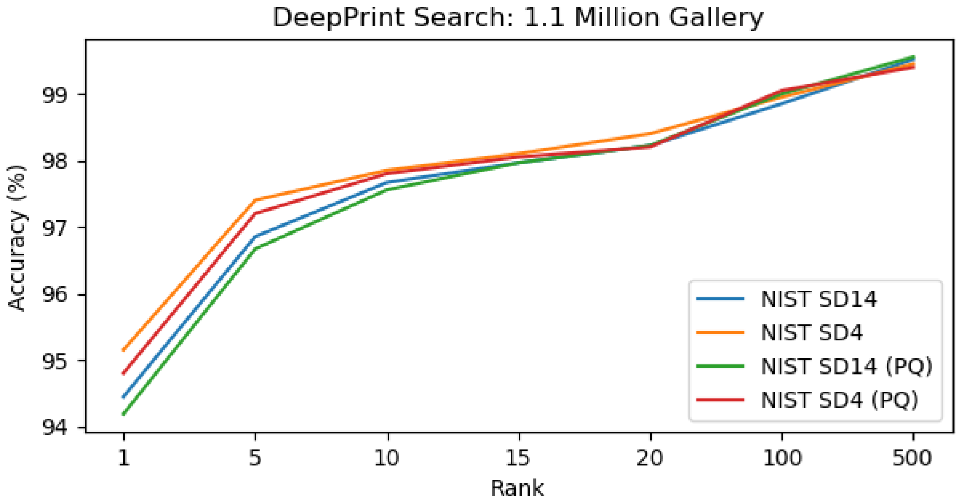


Figure 4.7 Closed-Set Identification Accuracy of DeepPrint (with and without Product Quantization (PQ)) on NIST SD4 and NIST SD14 (last 2,700 pairs) supplemented with a gallery of 1.1 Million. Rank-1 Identification accuracies are 95.15% and 94.44%, respectively. Search time is only 160 milliseconds. After adding product quantization, the search time is reduced to 51 milliseconds and the Rank-1 accuracies only drop to 94.8% and 94.2%, respectively.

4.10.2 Minutiae Re-ranking

Using the open-source minutiae matcher proposed in [20], COTS A and COTS B, we re-rank the top-500 candidates retrieved by the DeepPrint representation to further improve the Rank-1

Table 4.9 DeepPrint + Minutiae Re-ranking Search Accuracy (1.1 million background)

Metric	NIST SD4 Rank-1 Search Accuracy	NIST SD14 Rank-1 Search Accuracy	Search Time (milliseconds)[1]
DeepPrint + [20]	98.8%	98.22%	**300**
DeepPrint + COTS A[2]	**99.45**%	99.48%	11,000
DeepPrint + COTS B[2]	98.25%	98.41%	13,000
COTS A[3]	98.85%	**99.51**%	27,472
COTS B[3]	89.2%	85.6%	428

[1] Search times benchmarked on an Intel Core i9-7900X CPU @ 3.30GHz
[2] COTS only used for re-ranking the top 500 DeepPrint candidates.
[3] COTS used to perform search against the entire 1.1 million gallery.

Table 4.10 DeepPrint + PQ: Search Accuracy (1.1 million background)

Algorithm	NIST SD4 Rank 1 Search Accuracy	NIST SD14 Rank1 Search Accuracy	Search Time (milliseconds)[1]
DeepPrint	95.15%	94.44%	160
DeepPrint + PQ	94.80%	94.18%	**51**

[1] Search times benchmarked on an Intel Core i9-7900X CPU @ 3.30GHz

identification accuracy. Following this re-ranking, we obtain competitive search accuracy as the top COTS SDK, but at significantly faster speeds (Table 4.9).

4.10.3 Product Quantization

We further improve the already fast search speed enabled by the DeepPrint representation by performing product quantization on the templates stored in the gallery. This reduces the DeepPrint template size to only **64 bytes** and reduces the search speed down to **51** milliseconds from 160 milliseconds with only marginal loss to search accuracy (Table 4.10) and (Fig. 4.7).

Table 4.11 DeepPrint Representation Comparison

Metric	Minutiae Representation[1]	Texture Representation[1]	Fused Representation[2]
FVC 2004 DB1A TAR @ FAR = 0.1%	97.4%	90.0%	**97.5**%
NIST SD4 TAR @ FAR = 0.01%	97.0%	97.15%	**97.9**%
NIST SD14 TAR @ FAR = 0.01%	97.29%	98.14%	**98.55**%

[1] Each representation (96 bytes) is extracted from one branch in the DeepPrint architecture. [2] Scores from the minutiae representation are fused with the texture representation using sum score fusion.

Table 4.12 DeepPrint Ablation Study

Metric	w/o all	with alignment	with alignment + domain knowledge
FVC 2004 DB1A TAR @ FAR = 0.1%	72.86%	88.0%	**97.5%**
NIST SD4 TAR @ FAR = 0.1%	96.95%	96.65%	**97.9%**
NIST SD14 TAR @ FAR = 0.1%	97.96%	96.52%	**98.55%**

4.11 Ablation Study

Finally, we perform an ablation study to highlight the importance of (i) the automatic alignment module in the DeepPrint architecture and (ii) the minutiae-map domain knowledge added during training of the network. In our ablation study, we report the authentication performance of DeepPrint with/without the constituent modules.

We note that in all scenarios, the addition of domain knowledge improves authentication performance (Tables 4.11 and 5.6). This is especially true for the FVC 2004 DB1 A database which is comprised of slap fingerprints with different characteristics (size, distortion, conditions) than the rolled fingerprints used for training DeepPrint. Thus we show how adding minutiae domain knowledge enables better generalization of DeepPrint to datasets which are very disparate from its training dataset. We note that alignment does not help in the case of NIST SD4 and NIST SD14 (since rolled fingerprints are already mostly aligned), however, it significantly improves the performance on FVC 2004 DB1 A where fingerprint images are likely to be severely unaligned.

We also note that the minutiae-based representation and the texture-based representation from DeepPrint are indeed complementary, evidenced by the improvement in accuracy when fusing the scores from both representations. (Table 4.11).

4.12 Interpretability

As a final experiment, we demonstrate the interpretability of the DeepPrint representation using the deconvolutional network proposed in [189]. In particular, we show in Fig. 4.8 which pixels in an input fingerprint image are fixated upon by the DeepPrint network as it extracts a representation. From this figure, we make some interesting observations. In particular, we note that while the texture branch of the DeepPrint network seems to only focus on texture surrounding singularity points in the fingerprint (core points, deltas), the minutiae branch focuses on a larger portion of the fingerprint in areas where the density of minutiae points are high. This indicates to us that our guiding the DeepPrint network with minutiae domain knowledge does indeed draw the attention of the network to minutiae points. Since both branches focus on complementary areas and features, the fusion of the representations improves the overall matching performance (Table 4.11).

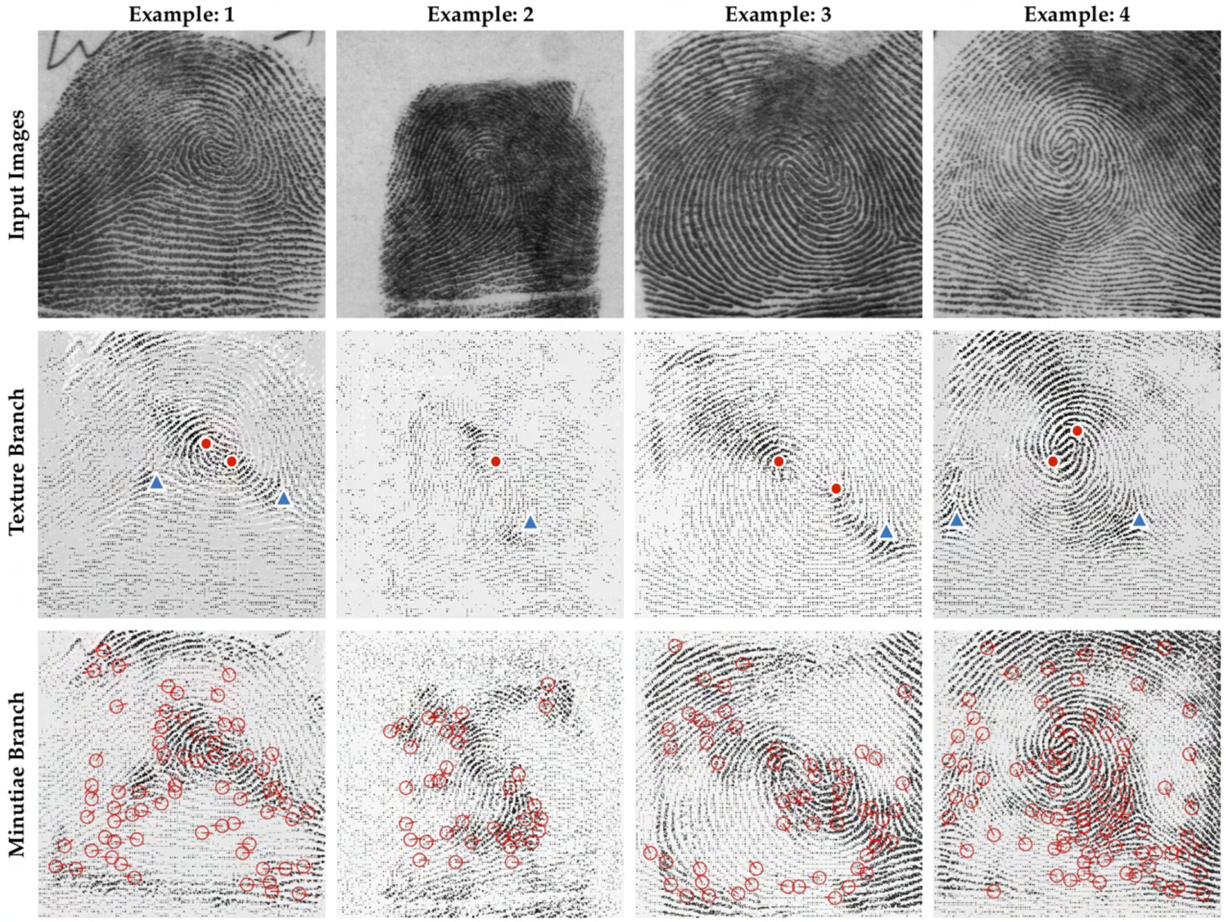


Figure 4.8 Illustration of DeepPrint interpretability. The first row shows three example fingerprints from NIST SD4 which act as inputs to DeepPrint. The second row shows which pixels the texture branch is focusing on as it extracts its feature representation. Singularity points are overlaid to show that the texture branch fixates primarily on regions surrounding the singularity points. The last row shows pixels which the minutiae branch focuses on as it extracts its feature representation. We overlay minutiae to show how the minutiae branch focuses primarily on regions surrounding minutiae points. Thus, each branch of DeepPrint extracts complementary features which comprise more accurate and interpretable fixed-length fingerprint representations than previously reported in the literature.

4.13 Computational Resources

DeepPrint models and training code are implemented in Tensorflow 1.14.0. All models were trained across 2 NVIDIA GeForce RTX 2080 Ti GPUs. All search and authentication experiments were performed with an Intel Core i9-7900X CPU @ 3.30GHz and 32 GB of RAM.

4.14 Summary

We have presented the design of a custom deep network architecture, called DeepPrint, capable of extracting highly discriminative fixed-length fingerprint representations (200 bytes) for both authentication (1:1 fingerprint comparison) and search (1:N fingerprint comparison). We showed how alignment and fingerprint domain knowledge could be added to the DeepPrint network architecture to significantly improve the discriminative power of its representations. Then, we benchmarked DeepPrint against two state-of-the-art COTS matchers on a gallery of 1.1 million fingerprints, and showed competitive search accuracy (DeepPrint Rank-1 of 98.8% vs. COTS 98.85% on NIST SD4) at significantly faster speeds (300 ms vs. 27,000 ms against a gallery of 1.1 million). We also showed how the DeepPrint representation could be used for matching in the encrypted domain via fully homomorphic encryption. We posit that the compact, fixed-length DeepPrint representation will significantly aid in large-scale fingerprint search. Among the three most popular biometric traits (face, fingerprint, and iris), fingerprint is the only modality for which no state-of-the-art fixed-length representation is available. This work aims to fill this void.

Chapter 5

Infant Fingerprint Recognition

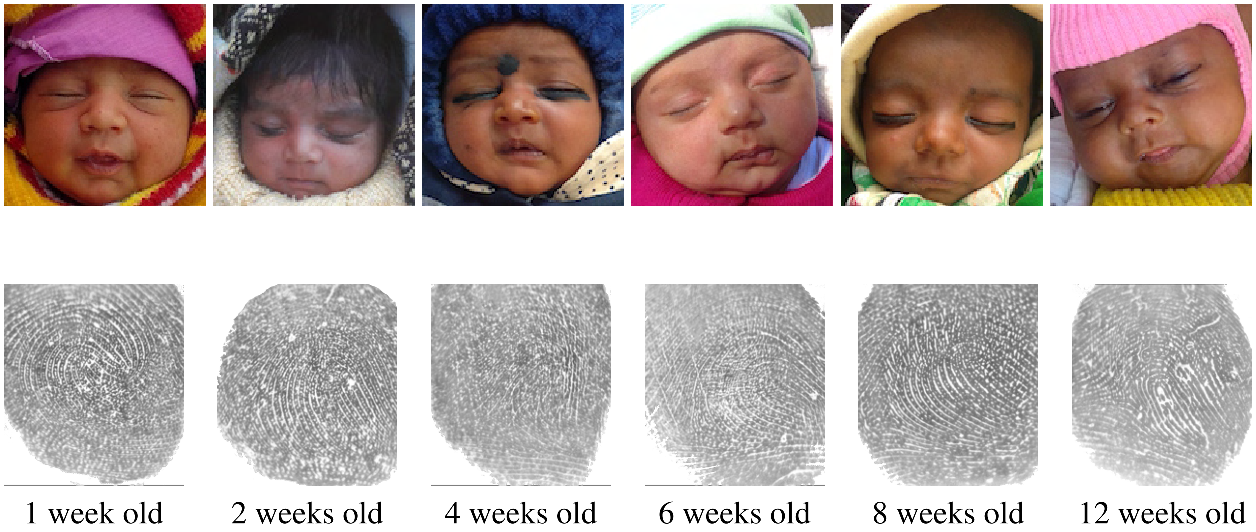


Figure 5.1 Face images (top row) and corresponding left thumb fingerprints (bottom row) of six different infants under 3 months of age. Face images were captured by a *Xiaomi MI A1* smartphone camera and fingerprint images were captured by our 1,900 ppi RaspiReader [49, 82] at the Saran Ashram Hospital, a charitable organization in Dayalbagh, Agra, India.

In the previous three chapters, we focused on improving each of the sub-modules of fingerprint recognition systems (fingerprint readers via the Universal Target and RaspiReader, and feature extractors and matchers via DeepPrint). In this final chapter, we look to extend fingerprint recognition to all ages with the goal of alleviating infant suffering and mortality around the world. In particular in many of the least developed and developing countries, a multitude of infants continue to suffer and die from vaccine-preventable diseases and malnutrition. Lamentably, the lack of official identification documentation makes it exceedingly difficult to track which infants have been vaccinated and which infants have received nutritional supplements. Answering these questions could prevent this infant suffering and premature death around the world. To that end, we propose Infant-Prints, an end-to-end, low-cost, infant fingerprint recognition system [49]. Infant-Prints is comprised of our (i) custom built, compact, low-cost (85 USD), high-resolution (1,900 ppi), ergonomic fingerprint reader, and (ii) high-resolution infant fingerprint matcher. To evaluate the efficacy of Infant-Prints, we collected a longitudinal infant fingerprint database captured in 4 different sessions over a 12-month time span (December 2018 to January 2020), from 315 infants at the Saran Ashram Hospital, a charitable hospital in Dayalbagh, Agra, India. Our experimental results demonstrate, for the first time, that Infant-Prints can deliver accurate and reliable recognition (over time) of infants enrolled between the ages of 2-3 months, in time for effective delivery of vaccinations, healthcare, and nutritional supplements (**TAR=95.2% @ FAR = 1.0%** for infants aged 8-16 weeks at enrollment and authenticated 3 months later).

5.1 Introduction

There are more than 600 million children living worldwide between the ages of 0-5 (years) [77] with an additional 353,000 more newborns setting foot on the planet each and every day [78]. A majority of these births take place in the poorest regions of the world, where it is likely that neither the infants nor their parents will have access to any official identification documents[1]. Even if the infant has obtained an official ID, it may be fraudulent or shared with others [180–182]. Without legitimate and verifiable identification, infants are often denied access to healthcare, immunization,

[1] Selecting and assigning a name to the newborns can be a drawn out process in developing countries in which parents consult immediate family members or even an astrologer for a proper name. While deciding upon a name, the infant is simply referred to as “baby” or “daughter of”, or “son of”.

and nutritional supplements. This is especially problematic for infants[2] (newborns to 12 months), given that they are at their most critical stage of development.

The downstream problems caused by lack of proper infant ID in the planet's least-developed countries can be quantitatively seen in the flat lining of global vaccination coverage. In particular, from 2015 to 2018, the percentage of children who have received their full course of three-dose diphtheria-tetanus-pertussis (DTP3) routine immunizations remains at about 85% [183]. This falls short of the GAVI Alliance (formerly Global Alliance for Vaccines and Immunization[3]) target of achieving global immunization coverage of 90% by 2020. According to UNICEF, 25 million children do not receive proper annual vaccination, leading to 1.5 million child deaths per annum from vaccine-preventable diseases[4]. The World Health Organization (WHO) suggests that inadequate monitoring and supervision and lack of official identification documents (making it exceedingly difficult to accurately track vaccination schedules) are key factors[5].

Infant identification is also urgently needed to effectively provide nutritional supplements. The World Food Program (WFP), a leading humanitarian organization fighting hunger worldwide, assists close to 100 million people in some of the poorest regions of the world[6]. However, often the food never reaches the intended beneficiaries because of fraud in the distribution system [180–182]. For example, the WFP found that in Yemen, a country with 12 million starving residents, food distribution records are falsified and relief is being given to people not entitled to it, preventing those who actually need aid from receiving it [180, 181].

Accurate and reliable infant recognition would also assist in baby swapping prevention[7], identifying missing or abducted children, and access to government benefits, healthcare, and financial services throughout an infant's lifetime.

[2] Infants are considered to be in the 0-12 months age range, whereas, toddlers and preschoolers are within 1-3 and 3-5 years of age, respectively [24].

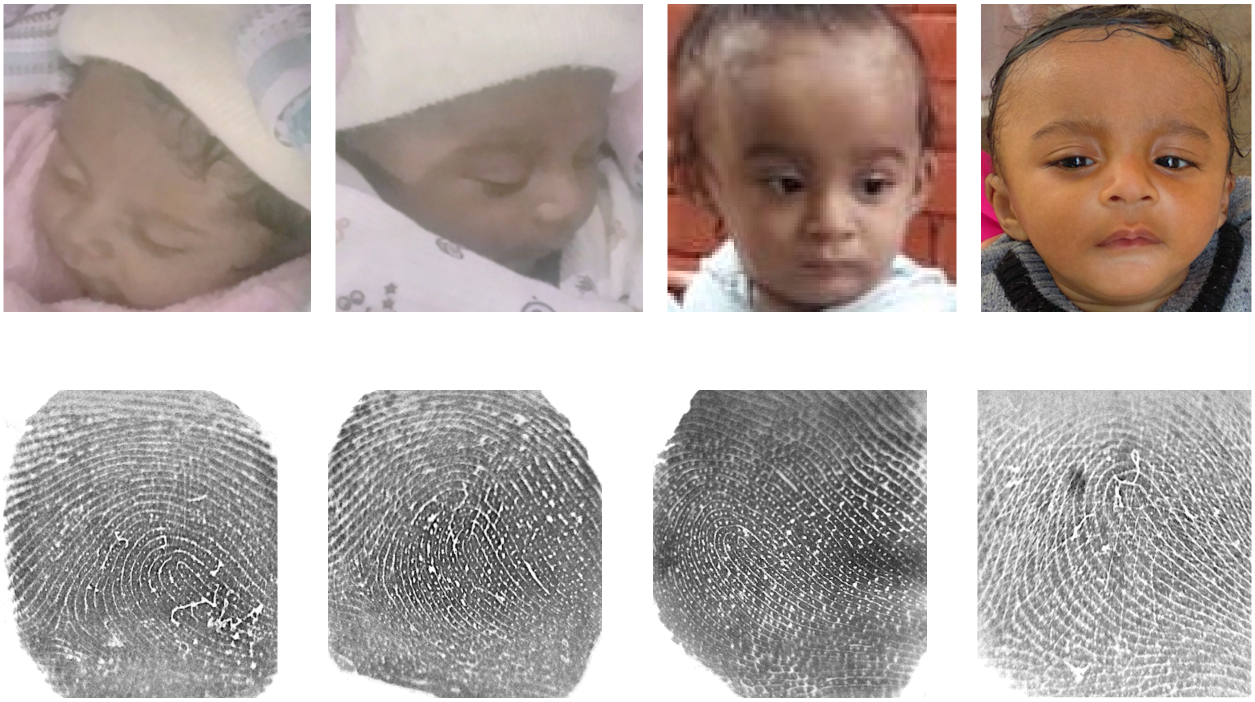
[3] https://bit.ly/1i7s8s2

[4] https://www.unicef.org/immunization

[5] https://bit.ly/1pWn6Gn

[6] https://evaw-un-inventory.unwomen.org/fr/agencies/wfp

[7] https://bit.ly/2U5eAHn



3 months 3 months, 2 days 6 months 12 months

Figure 5.2 Face images (top row) and corresponding left thumb fingerprints (bottom row) of an infant, *Meena Kumari*, acquired on (a) December 16, 2018 (Meena was 3 months old), (b) December 18, 2018 (3 months, 2 days old), (c) March 5, 2019 (6 months old), and (d) September 17, 2019 (12 months old) at Saran Ashram Hospital, Dayalbagh, India. Note that as Meena ages, fingerprint details emerge such as visible pores. This level of detail is enabled by our 1,900 ppi reader.

As we show in the next section, fingerprint recognition [111], is the only way to accurately and reliably establish an infant's identity. While fingerprint recognition is now a mature field and billions of teenagers and adults have been using it to authenticate themselves, children, particularly infants and toddlers, cannot yet utilize fingerprint recognition to get a unique and verifiable digital identity.

5.1.1 Fingerprints for Infant-ID

Conventional identification documents (paper records) are impractical for infant recognition in many of the least developed and developing countries because they are not securely linked to a specific infant. Furthermore, they may be fraudulent [182], lost, or stolen. We posit that a more accurate, robust, and verifiable means of infant recognition is through the use of *biometric*

Table 5.1 Related work on child and infant fingerprint recognition.

Study	Year	Resolution	# Subjects	Enrollment	Lapse
Galton [59]	1899	Inked	1	0 year	0 - 4.5 years
TNO [41]	2005	500 ppi	161	0 - 13 years	N/A*
BIODEV II [141]	2007	500 ppi	300	0 - 12 years	N/A*
UltraScan [149]	2006-2009	500 ppi	308	0 - 18 years	
Aadhar [68]	2009	500ppi	1.25B	5 years	N/A
JRC [97]	2013	500 ppi	2611	0 - 12 years	
Jain *et al.* [84]	2016	1,270 ppi	309	0 - 5 years	1 year
Saggese *et al.* [147][1]	2019	3,400 ppi	142	0 - 6 months	variable
Infant-Prints [82]	2019	1,900 ppi	194	0 - 3 mos.	3 mos.
Preciozzi *et al.* [138]	2020	500 ppi	16,865	0 - 18 years	10 years
This study	2020	1,900 ppi	315	0 - 3 months	1 year

* No time span available for these studies.

[1] Scores from across all time lapses (weeks or months) are aggregated when computing the fingerprint recognition error rates. This inflates the true longitudinal recognition performance.

recognition. Of the prominent biometric traits, we posit that fingerprint is the most promising for infant recognition. This is because, (i) face recognition is challenging due to the rapid aging of the infant's face from infanthood to childhood [173]. (ii) Iris recognition [11] is impractical because the infant will often be sleeping or crying. (iii) Footprint recognition [99, 104] requires removing socks and shoes and cleaning the infant's feet, and finally, (iv) palmprint recognition [101] requires opening an infant's entire hand where the concavity of the palm makes it difficult to image. In contrast, fingerprint recognition has already been shown to be practical for young children [84]. Furthermore, fingerprints have been shown to be (i) unique [136, 195], (ii) present at birth [7, 31, 133], (iii) stable over time in terms of recognition accuracy [85, 188], and (iv) a socially acceptable biometric trait to capture [84].

Fingerprint recognition of infants comes with its own challenges and requirements, including:

1. A compact, low-cost, ergonomic, high-resolution (to accommodate small inter-ridge spacings), and high throughput fingerprint reader.

2. A robust and accurate fingerprint matcher to accommodate low quality (distorted, dirty, wet, dry, motion blurred), high-resolution fingerprint images.

As such, prevailing COTS fingerprint recognition systems, designed primarily for an adult population, are not feasible for infant fingerprint recognition. Our goal then is to develop an end-to-end fingerprint recognition system, specifically designed for infants.

5.2 Related Work

Table 5.1 summarizes prior work on infant and child fingerprint recognition. These studies are summarized as follows:

- Beginning in 2004, the Netherlands Organization for Applied Scientific Research (TNO) conducted a study [41] wherein they concluded that it was not possible to obtain clear fingerprints from children under 4 years of age due to low fidelity in the ridge pattern on their fingers.
- A pilot program called BIODEV II was initiated in 2007 for capture, storage and verification of biometric data for Schengen visa applicants [141]. Experimental results based on the fingerprints of 300 children acquired in Damascus (Syria) and Ulan Bator (Mongolia), show that it is challenging to acquire fingerprints of children below 12 years of age.
- UltraScan conducted a study from 2006 to 2009 which modeled the growth of the fingerprints of children as they grow into their adolescence [149]. However, no experimental results were provided on child fingerprint capture and recognition.
- The Joint Research Center of the European Commission published a technical report [97] in 2013 on fingerprinting 2,611 children between 0 to 12 years of age. Fingerprints were acquired using 500 ppi fingerprint readers while passport processing by the Portuguese government. The report concluded that fingerprint recognition of children younger than 6 years of age is challenging.

- In 2016, Jain *et al.* acquired fingerprints of 309 children in the age range of 0 to 5 years via a 1,270 ppi fingerprint reader [84]. They concluded that it is feasible to recognize infants enrolled at the age of 6 months and authenticated one year later.

- In 2019, Saggese *et al.* acquired fingerprint images of 500 newborns and infants (less than 6 months of age) at the Tijuana General Hospital in Mexico using a custom built 3,400 ppi contactless fingerprint reader [147]. Although the authentication results reported seem promising, the study does not separate out the longitudinal recognition performance.

- Perciozzi *et al.* reported extremely low authentication performance of infants in a study published in 2020 [138]. The low performance can be attributed to the fact that the infant's fingerprints were captured with a standard 500 ppi fingerprint reader.

- In our preliminary study [82], we collected fingerprints of 194 infants via a custom 1,900 ppi fingerprint reader. We found that infants enrolled at ages 0-3 months can be accurately and reliably recognized 3 months later with TAR=90% @ FAR=0.1%.

Among the aforementioned studies, there are only three studies [82,138,147] which investigate the feasibility of recognizing infants under the age of 3 months at enrollment. (i) While the infant fingerprint recognition results reported in [147] by Saggese *et al.* seem promising, they aggregate scores from all time lapses (weeks or months) for computing the fingerprint recognition error rates which inflates the true longitudinal recognition performance. (ii) Preciozzi *et al.* report poor infant recognition results (TAR = 15.61% @ FAR=0.1% for 2-3 month old age group). (iii) Our preliminary study on infant fingerprint recognition [82] utilized a custom 1,900 ppi infant fingerprint reader, however, the matcher was not designed to fully utilize the high-resolution imagery (instead using existing matchers designed for 500 ppi images). Furthermore, the matcher did not incorporate any enhancement or aging of the friction ridge pattern. Finally, our preliminary study was conducted for 194 infants across a maximum time lapse of 3 months. In contrast, the completed work in this thesis includes 315 infants with longitudinal data of up to a **one year time lapse**.

The key differences between the present and prior work (specifically targeting infant recognition [82, 138, 147]) can be concisely summarized as follows:

- The longitudinal infant authentication and search performance has not been adequately addressed in prior works. In [147], fingerprint pairs captured across time lapses of different duration were lumped into the same evaluation. In our preliminary study [82], we only assessed the longitudinal performance for a time lapse of 3-months. In completing this thesis, we extend this longitudinal evaluation out to a full 12 month time lapse (requiring further in-situ data collection).

- Prior work proposed high-resolution fingerprint readers, but did not exploit the high-resolution imagery. Instead, the existing works utilize 500 ppi fingerprint matchers (designed for the adult population). In this thesis, we design a high-resolution fingerprint matcher specifically for infants to further improve the matching performance. Extensive ablation studies show the impact of these algorithmic improvements.

- This is the first comprehensive study to develop an **entire**, end-to-end infant fingerprint recognition system (including fingerprint reader, matcher, and mobile application (Fig. 5.3)), and then rigorously evaluate the system on a longitudinal, in-situ dataset to successfully demonstrate that infants can be enrolled at ages of less than 3 months, and then recognized after a time lapse of 12 months with acceptable accuracy. The study in this thesis is more

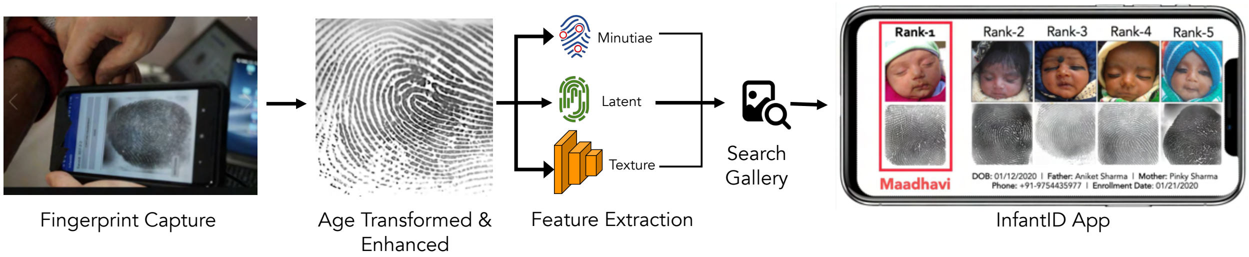

Fingerprint Capture
Age Transformed & Enhanced
Minutiae
Latent
Texture
Feature Extraction
Search Gallery
Rank-1
Rank-2
Rank-3
Rank-4
Rank-5
Maadhavi
DOB: 01/12/2020 | Father: Aniket Sharma | Mother: Pinky Sharma
Phone: +91-9754435977 | Enrollment Date: 01/21/2020
InfantID App

Figure 5.3 Overview of the Infant-Prints system.

The specific technical contributions of our approach are as follows:

- Design and prototyping of a compact (1”×2”×3”), low-cost (85 USD), high-resolution (1,900 ppi), ergonomic fingerprint reader for infants (Fig. 5.4). This reader is much smaller and better designed for infants than our earlier open sourced fingerprint reader proposed in [47]. We also prototype a contactless version of our fingerprint reader (Fig. 5.6) in order to compare contact-based sensing technologies with contactless sensing technologies when used for infants.

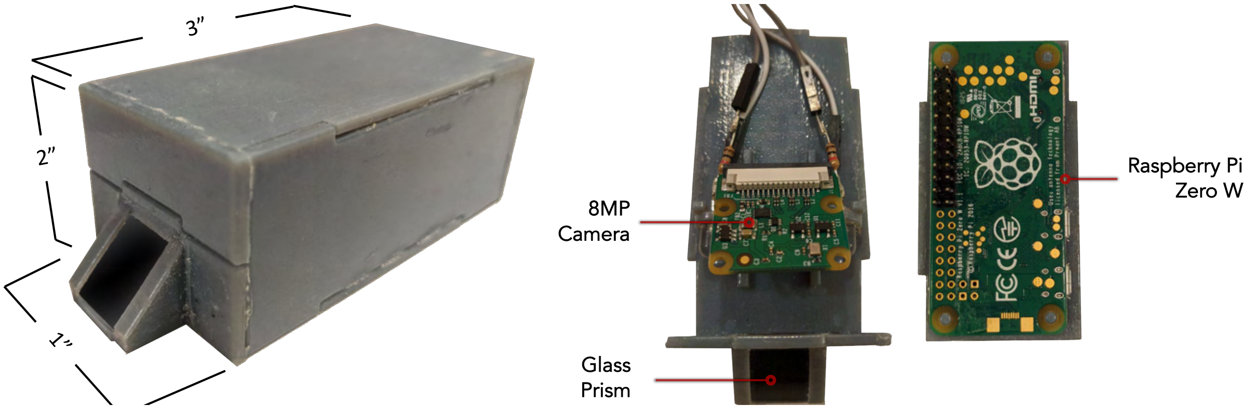
- Collection of a longitudinal infant fingerprint database comprised of 315 infants (0-3 months) over 4 separate sessions separated by 13 months (between December 2018 and January 2020). The data was collected at the Saran Ashram hospital, Dayalbagh, India.

- A first-of-its-kind, high resolution fingerprint matcher for infants which incorporates infant fingerprint aging and enhancement modules together with high resolution texture and minutiae matchers.

- The experimental results evaluated on our longitudinal infant dataset indicate that indeed, it is possible to enroll infants at ages younger than 3 months and accurately recognize them months later based only upon their fingerprints **TAR=95.2%@FAR=1.0%,TAR=92.8%@FAR=0.1%** (for infants enrolled at 2-3 months of age, and authenticated 3 months later), **TAR=85%@FAR=1.0%** for infants enrolled at 2-3 months of age, and authenticated a full year later.

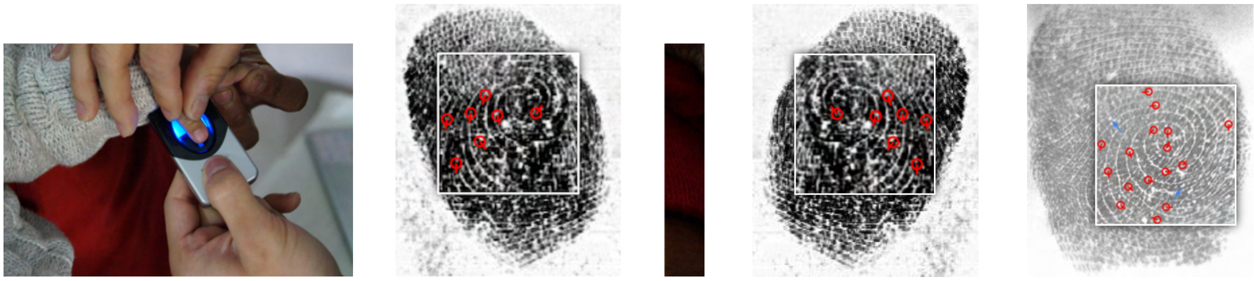
5.3 High-Resolution Fingerprint Reader

Almost all the fingerprint readers used in government and commercial applications capture images at a resolution of 500 ppi. This resolution is sufficient to resolve adult fingerprint ridges that have an inter-ridge spacing of about 8-10 pixels. However, 500 ppi resolution is not adequate for infant



(a) (b)

Figure 5.4 Prototype of the 1,900 ppi compact (1” × 2” × 3”), ergonomic fingerprint reader. An infant’s finger is placed on the glass prism with the operator applying slight pressure on the finger. The capture time is 500 milliseconds. The prototype can be assembled in less than 2 hours. See the video at: https://www.youtube.com/watch?v=f8tYbE9Cwd0.



(a) U.are.U 4500 (b) 500 ppi (c) RaspiReader (d) 1,900 ppi

Figure 5.5 An infant’s fingerprints are acquired via (a) a 500 ppi commercial reader (Digital Persona U.are.U 4500) and (c) our custom 1,900 RaspiReader. The captured fingerprint images of the right thumb from the commercial reader and the Infant-Prints reader for a 13 day old infant are shown in (b) and (d), respectively. Manually annotated minutiae are shown in red circles (location) with a tail (orientation). Blue arrows denote pores on the ridges.

fingerprint capture since infant fingerprints have an inter-ridge spacing of 4-5 pixels (sometimes the width of a valley may be less than 1 pixel for an infant fingerprint captured at 500 ppi).

Some cheaper readers (50 USD) reach 1,000 ppi only after upsampling the fingerprint image [155]. However, Jain *et al.* [84] showed that even at a native resolution[8] of 1,270 ppi, fingerprint recognition of young infants (0-6 months) was much lower than infants 6 months and older. The lack of an affordable, compact and high resolution fingerprint reader motivated us to construct

[8] Native resolution is the resolution at which the sensor is capable of capturing (no upsampling or downsampling).

a first-of-a-kind, 1,900 ppi fingerprint reader, called RaspiReader (Fig. 5.4), enabling capture of high-fidelity infant fingerprints (Fig. 5.5), particularly those in the age range 0-3 months. Unlike our prior efforts to build a compact and cheap reader for adults [44,47], both the cost and size of the infant fingerprint reader has been significantly reduced (from 180 USD to 85 USD and $4" \times 4" \times 4"$ to $1" \times 2" \times 3"$). Furthermore, the fingerprint reader is now more ergonomic for infant fingerprints since it has a glass prism towards the front of the reader (Fig. 5.4) rather than flush with the top of the reader (as is the case with commercial readers). Since infants frequently clench their fists and have very short fingers, placing the prism out front significantly eases placement of an infant's finger on the platen (Fig. 5.5 (b)).

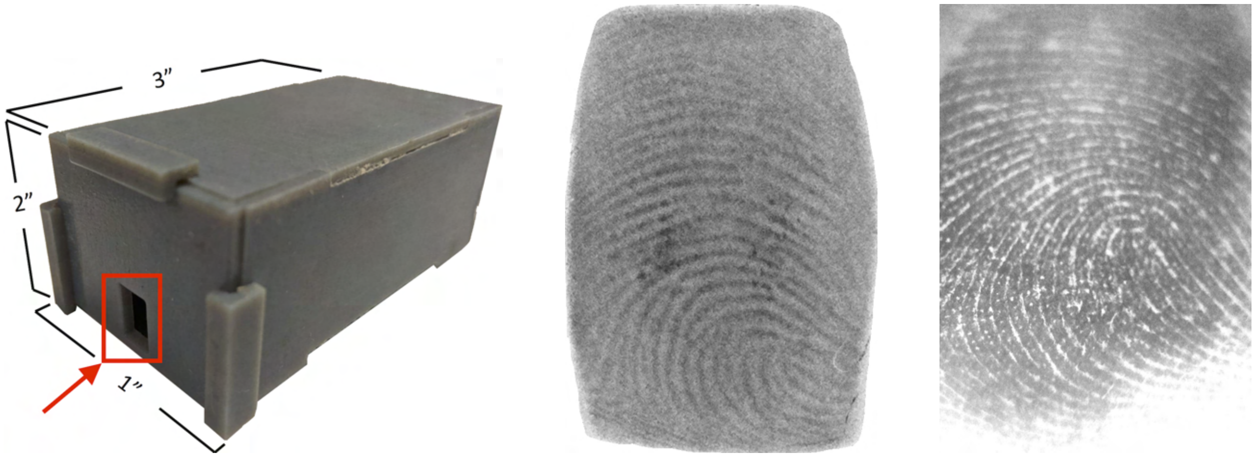
The entire design and 3D parts for the reader casing along with step by step assembly instructions are open sourced.[9] Figure 5.5 shows that this custom 1,900 ppi fingerprint reader is able to capture (500 millisecond capture time) the minute friction ridge pattern of a 13 day old infant (both minutiae and pores) with higher fidelity than the 500 ppi Digital Persona U.are.U. 4500 reader.

We also prototype a contactless variant of our contact-based infant fingerprint reader. Similar to [147], we adopt a different size finger rest for different size thumbs. In this manner, we are able to compare contact-based high resolution fingerprint readers with the high resolution contactless sensing technology. Figure 5.6 shows an example infant fingerprint captured by both our contactless and contact-based fingerprint reader.

5.4 Longitudinal Fingerprint Dataset

To effectively demonstrate the utility of an infant fingerprint recognition system for the applications we have highlighted above, we must be able to show its ability to recognize a child based on fingerprints acquired months after the initial enrollment. Such an evaluation requires a longitudinal fingerprint dataset which contains fingerprint images of the same infant over time at successive intervals. Collecting such a dataset is a significant challenge as it requires the cooperation of

[9] https://github.com/engelsjo/RaspiReader



(a) (b) Contactless image (c) Contact-based image

Figure 5.6 (a) Prototype of our 1,900 ppi contactless fingerprint reader. During capture, an infant's finger is placed on top of a small, contactless, rectangular opening (annotated in red) on the reader (the size of this opening can be adjusted with different sized slots). A camera captures the infant's fingerprint from behind the rectangular opening. Examples of a processed (segmented, contrast enhanced), contactless infant thumb-print (2 months old) is shown in (b) and the same infant's thumb-print acquired via contact-based fingerprint reader in (c).

an infant's parents in returning to the clinic multiple times for participation in the study. It also requires working with uncooperative infants who may become hungry or agitated during the data collection (our ergonomic fingerprint reader alleviated some of these challenges).

We have collected a dataset comprised of longitudinal fingerprint images of 315 infants (all enrolled at 0-3 months of age) at the Saran Ashram hospital in Dayalbagh, India across four sessions (see Fig. 5.7)[10]:

1. Session 1: December 12-19, 2018
2. Session 2: March 3-9, 2019
3. Session 3: September 12-21, 2019
4. Session 4: January 17-24, 2020

[10] Our dataset collection was approved by the Institutional Review Board (IRB) of Michigan State University and ethics committee of Dayalbagh Educational Institute and Saran Ashram Hospital. The fingerprint dataset cannot be made publicly available per the IRB regulations and parental consents.

Table 5.2 Infant Longitudinal Fingerprint Dataset Statistics

# Sessions	4
# Infants	315
Total # images	3,071
Age at enrollment	0 - 3 mos.
# Subjects with no time lapse*	127
# Subjects with 3 months lapse*	121
# Subjects with 6 months lapse*	29
# Subjects with 9 months lapse*	101
# Subjects with 12 months lapse*	41
Male to Female Ratio	43% to 57%

* Time lapse between enrollment and authentication image.

The infants were patients of the pediatrician, Dr. Anjoo Bhatnagar (Fig. 5.7). Prior to data collection, the parents were required to sign a consent form (approved by authors' institutional review board and the ethics committee of Saran Ashram hospital).

In a single session, we attempted to acquire a total of two impressions per thumb (sometimes we captured more (*e.g.* 4 impressions) or less (*e.g.* 1 impression) depending on the cooperative nature of the infant). Although a modest incentive was offered to parents for their data collection efforts, it was often difficult for them to meet our fingerprint capture schedule because of festivals, vacations, moving to a different city or loss of interest in the project. For this reason, out of the 315 total infants that we encountered, 25 infants were present in all four sessions, 54 infants came to only three sessions, 109 infants came to only two sessions, and 127 infants came to only one session. During collection, a dry or wet wipe was used, as needed, to clean the infant's finger prior to fingerprint acquisition. On average, data capture time, for 4 fingerprint images (2 per thumb) and a face image per infant, was 3 minutes[11]. This enabled a reasonably high throughput during the in-situ evaluation, akin to the operational scenario in immunization and nutrition distribution centers. Longitudinal fingerprint dataset statistics are given in Table 5.2.

[11] Data capture time includes parents signing the consent forms, record-keeping, and pacifying non-cooperative infants.

Figure 5.7 Infant fingerprint collection at Saran Ashram hospital, Dayalbagh, India. Pediatrician, Dr. Anjoo Bhatnagar, explaining longitudinal fingerprint study to the mothers while the authors are acquiring an infant's fingerprints in her clinic. Parents also sign a consent form approved by the Institutional Review Board (IRB) of our organizations.

5.5 Infant Fingerprint Matching

State-of-the-art fingerprint feature extractors and matchers are designed to operate on 500 ppi adult fingerprint images. This limitation forced the authors in [84] to down-sample the fingerprint images captured at 1,270 ppi to enable compatibility with COTS (Commercial Off The Shelf) matchers. The authors in [147] also had to down-sample images captured at 3,400 ppi in order to make them compatible with adult fingerprint matching systems. In our preliminary study [82], we developed a custom Convolutional Neural Network (CNN) based texture-matcher which directly operates on 1,900 ppi fingerprint images so that we did not have to down-sample images and discard valuable discriminative cues available in high resolution images. The final matching score in [82] was based on the fusion of (i) our CNN-based custom texture matcher and (ii) two state-of-the-art COTS matchers.

In completing this thesis, we (i) incorporate an enhancement and fingerprint aging preprocessing module, (ii) improve our high-resolution texture matcher from [82], and (iii) propose a high-resolution minutiae extractor trained on manually annotated infant fingerprint images. Combining these algorithmic improvements with two state-of-the-art fingerprint matchers (a latent fingerprint matcher, and a minutiae matcher) enables us to improve our recognition accuracy over that which was reported in our preliminary study [82]. In the following subsections, we discuss in more detail each of these algorithmic improvements.

5.5.1 Minutiae Matcher

Our high resolution minutiae matcher is comprised of (i) a high-resolution minutiae extractor, (ii) a minutiae aging model, and (iii) the Verifinger v11.0 ISO minutiae matcher. In the following subsections, we describe each of these algorithmic components.

5.5.2 Minutiae Extraction

Recent approaches to minutiae extraction in the literature have found that deep networks are capable of delivering superior minutiae extraction performance in comparison to traditional approaches [125, 126, 168, 191]. Furthermore, the authors in [20] showed that deep learning based minutiae extractors are particularly well suited for low quality fingerprint images. Since infant fingerprints can also be regarded as a "low-quality" fingerprint (heavy non-linear distortion, motion blur from uncooperative subjects, small inter-ridge spacing, very moist or dry fingers, dirty fingers), we choose to adopt the deep learning based minutiae extraction approach from [20] (with modifications to the architecture and training procedure) for high-resolution infant minutiae extraction. In our experiments, we demonstrate that the high-resolution minutiae extractor is capable of boosting the infant fingerprint recognition performance.

The core of the minutiae extraction algorithm proposed in [20] is a fully-convolutional autoencoder $M(.)$ which is trained to regress from an input fingerprint image $\mathbf{I} \in \mathbb{R}^{n \times m}$ to a ground truth minutiae map $\mathbf{H} \in \mathbb{R}^{n \times m \times 12}$ via $\hat{\mathbf{H}} = M(I)$, where $\hat{\mathbf{H}}$ is the predicted minutiae map. The spatial locations of hot spots in the minutiae map indicate the locations of minutiae points, and the 12 different channels of the minutiae map encode the orientation of the minutiae points. The parameters of M are trained in accordance with Equation (5.5.1).

$$\mathcal{L}_{minutiae} = ||\hat{\mathbf{H}} - \mathbf{H}||_2^2 \tag{5.5.1}$$

This estimated 12-channel minutiae map $\hat{\mathbf{H}}$ can be subsequently converted into a variable length minutiae set $\{(x_1, y_1, \theta_1), ..., (x_N, y_N, \theta_N)\}$ with N minutiae points via an algorithm which

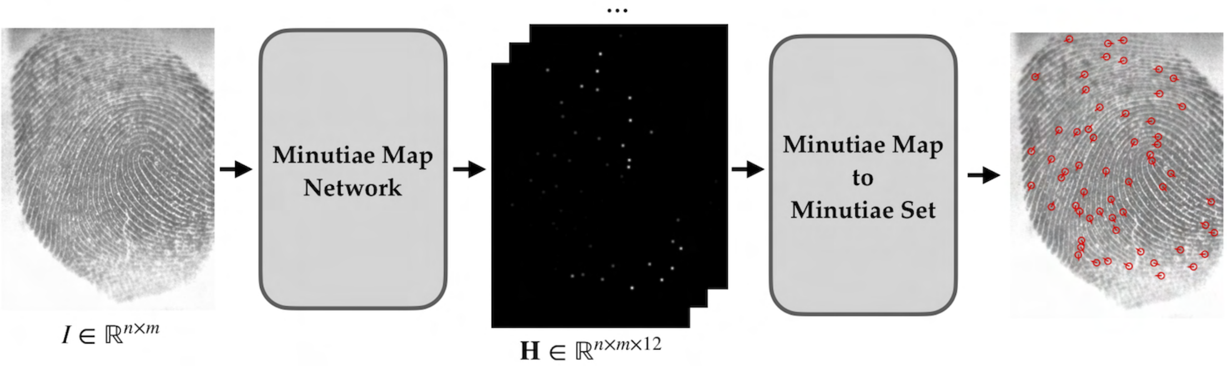

Minutiae Map Network
...
Minutiae Map to Minutiae Set
$I \in \mathbb{R}^{n \times m}$
$\mathbf{H} \in \mathbb{R}^{n \times m \times 12}$

Figure 5.8 Overview of the minutiae extraction algorithm. An input fingerprint of any size ($n \times m$) is passed to the minutiae extraction network (Table 5.3). The network outputs a $n \times m \times 12$ minutiae map $\mathbf{H}$ which encodes the minutiae locations and orientations of the input fingerprint. Finally, the minutiae map is converted to a minutiae set $\{(x_1, y_1, \theta_1), ..., (x_N, y_N, \theta_N)\}$ of N minutiae.

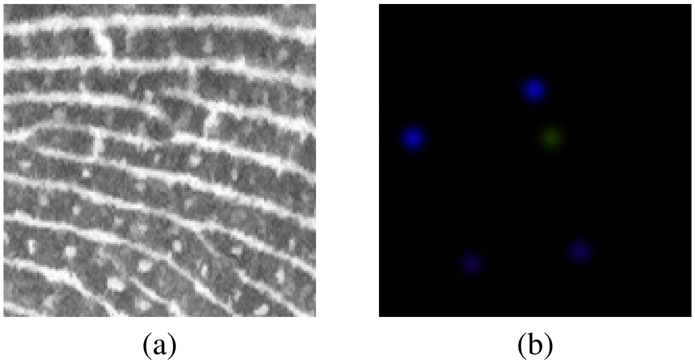


Figure 5.9 An example infant fingerprint patch (a) and the corresponding minutiae map (b). Note, we only show 3 channels of the 12 channel minutiae map here for illustrative purposes (red channel is the first channel, green is the fifth channel, and blue is the ninth channel). Given the full 12 channels of the minutiae map in (b), we can compute the minutiae locations (x, y) and orientations θ of the 1,900 ppi fingerprint patch in (a).

locates local maximums in the channels (locations) and individual channel contributions (orientations) followed by non-maximal suppression to remove spurious minutiae [20].

To obtain ground truth minutiae maps $\mathbf{H}$ for computing $\mathcal{L}_{minutiae}$, we encode a ground truth minutiae set T for a given infant fingerprint into $\mathbf{H}$ following the approach of [20] for latent fingerprints.

An example infant fingerprint patch, and a few channels of its 12 channel ground truth minutiae map are shown in Figure 5.9. An overview of our end-to-end minutiae extraction algorithm is shown in Figure 5.8. In contrast to the 500 ppi latent fingerprint minutiae extractor in [20], we

Table 5.3 Minutiae Extraction Network

Type	Output Size	Filter Size, Stride
Convolution	$256 \times 256 \times 64$	$4 \times 4, 1$
Convolution	$128 \times 128 \times 64$	$4 \times 4, 2$
Convolution	$64 \times 64 \times 128$	$4 \times 4, 2$
Convolution	$32 \times 32 \times 256$	$4 \times 4, 2$
Convolution	$16 \times 16 \times 384$	$4 \times 4, 2$
Convolution	$8 \times 8 \times 512$	$4 \times 4, 2$
Convolution	$8 \times 8 \times 1024$	$4 \times 4, 1$
Convolution	$4 \times 4 \times 1024$	$4 \times 4, 2$
Deconvolution	$4 \times 4 \times 1024$	$4 \times 4, 1$
Deconvolution	$8 \times 8 \times 512$	$4 \times 4, 2$
Deconvolution	$16 \times 16 \times 384$	$4 \times 4, 2$
Deconvolution	$32 \times 32 \times 256$	$4 \times 4, 2$
Deconvolution	$64 \times 64 \times 128$	$4 \times 4, 2$
Deconvolution	$128 \times 128 \times 64$	$4 \times 4, 2$
Deconvolution	$256 \times 256 \times 32$	$4 \times 4, 2$
Deconvolution	$256 \times 256 \times 12$	$4 \times 4, 1$

[†] During training, input patches are 256×256. During testing, the input can be of any size (the network is fully convolutional).

directly train our minutiae extractor on infant fingerprint patches at 1,900 ppi resolution. In this manner, we do not remove any discriminative cues (via down-sampling) from the input infant fingerprint images prior to performing minutiae extraction. Operating at a high resolution requires a deeper network architecture than that which was utilized in [20]. Our network architecture is shown in detail in Table 5.3. Note that while we train our auto-encoder on infant fingerprint *patches*, during test time, we input a full size infant fingerprint (of varying width and height) since our architecture is *fully-convolutional* and as such, is amenable to different size inputs.

5.5.2.1 Manual Minutiae Markup for Training

As seen in the previous section, obtaining ground truth minutiae maps $\mathbf{H}$ for training our minutiae map extraction network $M(.)$ requires a ground truth minutiae set T for each input infant fingerprint. To obtain these ground truth minutiae sets for training, we manually annotate the minutiae locations and orientations of 610 infant fingerprints in our dataset for which we had limited longitudinal data (*i.e.* the infant only visited 1 or 2 sessions). These fingerprints are separated from our evaluation dataset. We manually annotated the infant fingerprints using the GUI tool shown in Figure 5.10. The tool enables the addition of new minutiae and the removal of spurious minutiae. To make the markup task easier, we first automatically annotate the minutiae points on the 610 infant fingerprints using the Verifinger v11.0 minutiae extraction SDK. Then, we manually refine the Verifinger annotations with our markup GUI. Each manually annotated fingerprint was reviewed multiple times by one of 4 experts in the field of fingerprint recognition.

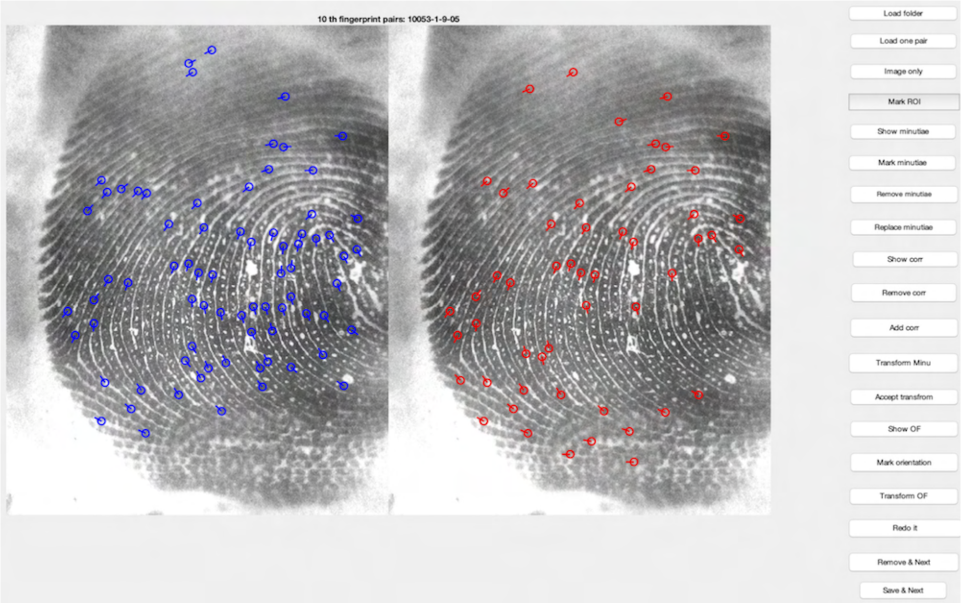


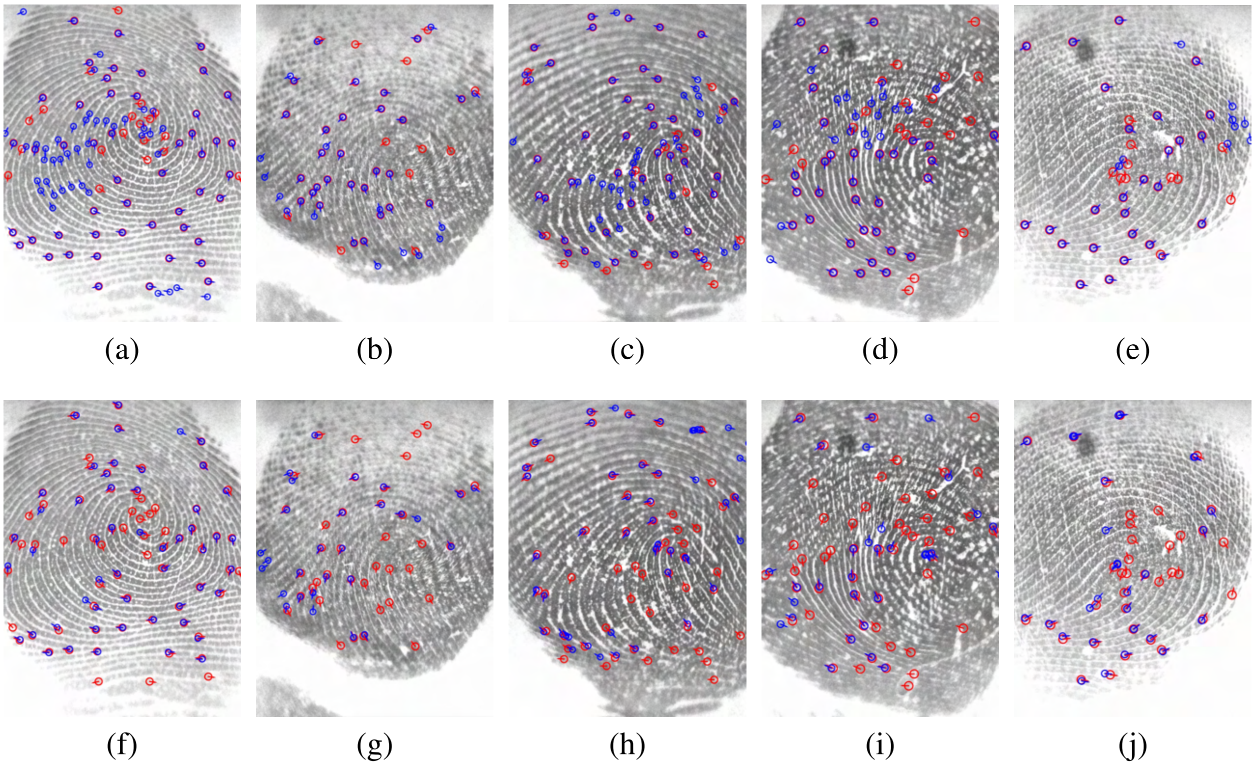
Figure 5.10 View of the manual minutiae markup/editing software used to markup minutiae locations on a subset of infant fingerprint images. These markups were later used as ground truth to train our high resolution infant minutiae extractor. The fingerprint on the left (blue annotations) is coarsely annotated with Verifinger v11 SDK to help speed up the annotation process. The fingerprint on the right (red annotations) shows the manually edited minutiae.

While the 610 manually annotated infant fingerprints provide an accurate ground truth dataset for training our minutiae extraction network, it is still small for training a deep network (Table 5.3). Therefore, rather than training our minutiae extraction network from scratch on the 610 manually annotated infant fingerprints, we first pretrain our minutiae extraction network on $9,508$ infant/child fingerprints collected in [85] and coarsely annotated with minutiae using the Verifinger v11.0 minutiae extractor. After pretraining our minutiae extraction network on these $9,508$ coarsely annotated (using Verifinger) fingerprints, we finally fine-tune all parameters of our network (Table 5.3) using our more accurate 610 manually annotated ground truth infant fingerprint images (560 used for training, 50 used for validation). We optimize our network parameters using the Adam optimizer and weight decay set to 4×10^{-5}. When training the network on the $9,508$ coarsely annotated training data, we use a learning rate of 0.01. When fine-tuning our network (all parameters fine-tuned) on our manually annotated fingerprint images, we reduce the learning rate to 0.0001. We use the minutiae detection accuracy on our 50 manually annotated validation fingerprints as a stopping criteria for the training. Finally, our network is trained on 256×256 patches to increase the number of training samples, and we employ data augmentations such as random rotations, cropping, translations, and flipping.

The efficacy of our high-resolution minutiae extraction algorithm is shown in Fig. 5.11. In comparison to Verifinger, our algorithm extracts significantly fewer spurious minutiae, while detecting nearly all of the true minutiae locations. We show in subsequent experiments that this results in a boost in infant fingerprint recognition performance.

5.5.2.2 Minutiae Aging

After extracting a minutiae set from an infant fingerprint with our high-resolution minutiae extractor, we further process the minutiae set via a minutiae aging model (Fig. 5.12). The authors in [138] showed that by linearly scaling an infant's fingerprint image, it could be better matched to an older fingerprint impression of the same infant. Note, that although the aging model in [138]


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

(f) (g) (h) (i) (j)

Figure 5.11 **Top row:** Verifinger minutiae detections; **Bottom row:** Minutiae detections from our high-resolution minutiae extractor. Manually marked minutiae are annotated in red. Note that Verifinger detects many of the true minutiae, but also extracts a significant number of spurious minutiae. Our proposed minutiae extractor has slightly lower detection accuracy (of true minutiae) than Verifinger, however, it extracts significantly fewer spurious minutiae. We further compare the two approaches quantitatively in our experimental results.

was shown to be beneficial for infant recognition, it did not result in desired levels of recognition accuracy due in part to the fact that the infant fingerprint images were captured at 500 ppi.

Rather than scaling an infant's fingerprint *image* as was done in [138], we directly scale the already extracted *minutiae set*. More formally, given a scale factor λ and a minutiae set T of N minutiae, where $T = \{(x_1, y_1, \theta_1), ..., (x_N, y_N, \theta_N)\}$, our scaled minutiae set $\hat{T}$ is given by:

$$\hat{T} = \{(\lambda x_1, \lambda y_1, \theta_1), ..., (\lambda x_N, \lambda y_N, \theta_N)\} \tag{5.5.2}$$

To determine the scale factor λ at which an infant's fingerprint pattern grows as they age, we select 82 pairs of our 610 manually annotated infant fingerprints for which we have longitudinal

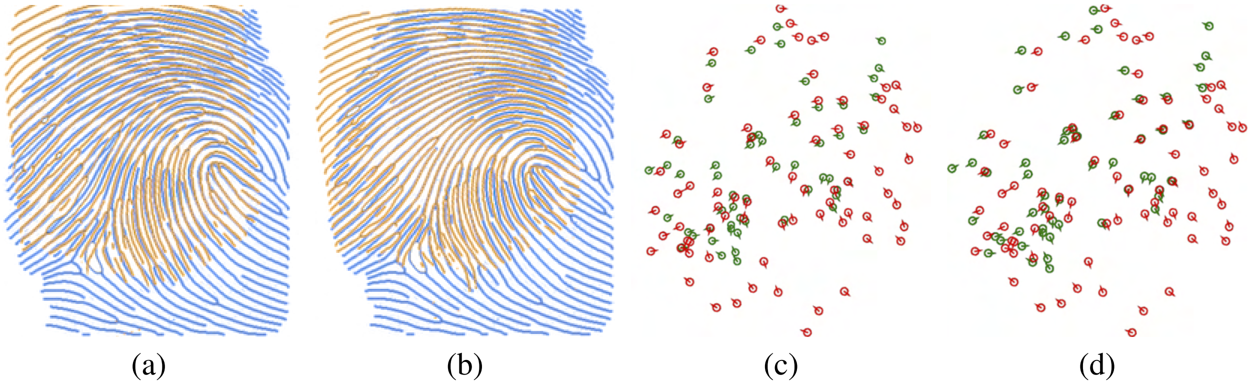


Figure 5.12 Effects of aging. (a) Acquired 3 month old enrollment image (orange) is overlaid on a 1 year old probe image (blue). (b) An aged 3 month old enrollment image (orange) is overlaid on a 1 year old probe image (blue). (c) 3 month old enrollment minutiae set (green) is overlaid on a 1 year old probe minutiae set (red). (d) An aged 3 month old enrollment minutiae set (green) is overlaid on a 1 year old probe minutiae set (red). Following aging (b, d), the enrollment image and probe image (and corresponding minutiae sets) overlap better.

impressions. The range of the time lapse ΔT (in weeks) for these 82 pairs of fingerprints is $12 \leq \Delta T \leq 40$ (mean $\Delta T = 34.3 \pm 10.3$). We then empirically evaluated different scalar factors in increments of 0.05 such that the minutiae matching accuracy (as computed by Verifinger v11 SDK) on these validation images was maximized. We found that applying a scalar factor of $\lambda = 1.1$ to infant images enrolled at less than 3 months provided the best recognition performance.

We also tried an adaptive aging model where the scalar factor was dependent upon the enrollment age and the elapsed time, but found no improvement in performance (likely because the majority age group in our experiments is infants enrolled between 2-3 months and recognized 3 months later, where the simple scalar value of $\lambda = 1.1$ suffices). Given similar performance, we kept the simpler static scalar aging model as opposed to the adaptive aging model.

An example of an infant minutiae set T and its corresponding aged minutiae set $\hat{T}$ is shown in Figure 5.12. In our experiments, we quantitatively demonstrate that this scaling of the enrollment minutiae points provides a boost to our recognition performance.

5.5.2.3 Minutiae Match Score

After extracting a minutiae set T (via our high-resolution minutiae extractor) and aging T into $\hat{T}$, we compute a minutiae matching score s_m between a probe infant fingerprint and an enrolled infant fingerprint using the Verifinger v11 ISO minutiae matcher.

5.5.3 Texture Matcher

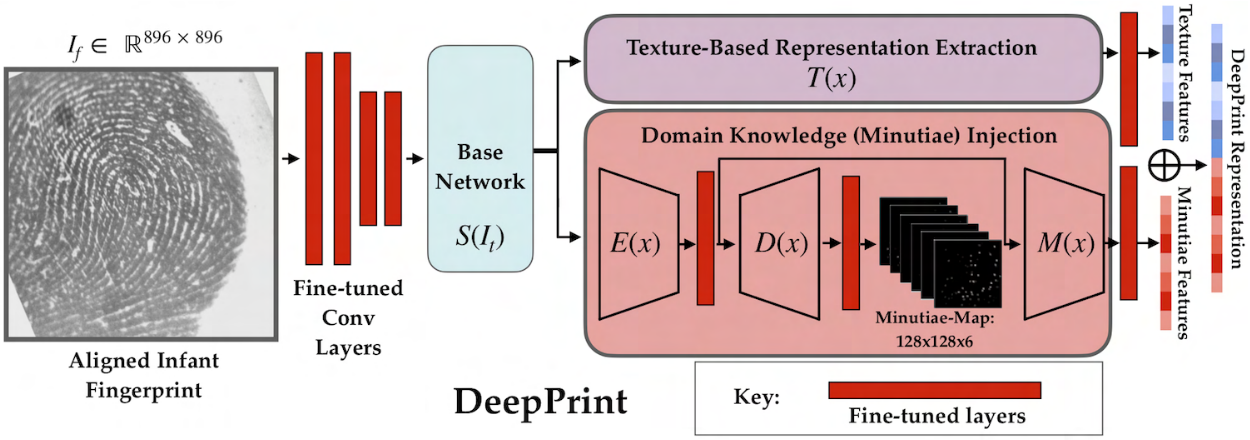


Figure 5.13 Overview of the Infant-Prints texture matcher. We modify DeepPrint [48] to accept 1,900 ppi high resolution infant fingerprint images. The network is pretrained on adult fingerprint images and then fine-tuned (red layers) with the infant dataset collected in [84].

Similar to latent fingerprints, infant fingerprints are often of poor quality and as such are difficult to accurately extract minutiae from (even with our high resolution minutiae extractor). Therefore, in addition to a minutiae match score, we also incorporate a texture matching score s_t using a state-of-the-art texture fingerprint matcher [48] [12]. Engelsma *et al.* [48] proposed a CNN architecture, called DeepPrint, embedded with fingerprint domain knowledge for extracting discriminative fixed-length fingerprint representations. Inspired by the success of DeepPrint to learn additional textural cues that go beyond just minutiae points, we adopt this matcher for infant fingerprint recognition. In particular, we modify the DeepPrint network architecture as follows: (i) the input size of 448×448 is increased to 1024×1024 (through the addition of convolutional layers) to

[12] Although DeepPrint also incorporates minutiae domain knowledge into the fixed-length representation, we refer to it as a texture matcher since minutiae points are not explicitly used for matching.

support 1,900 ppi images and (ii) the parameters of the added convolutional layers and the last fully connected layer are re-trained on the 1,270 ppi (upsampled to 1,900 ppi) longitudinal infant fingerprints acquired by Jain *et al.* in [84] combined with 610 of our 1,900 ppi images which we set aside for training. In total, we re-train the network with 9,683 infant fingerprint images from 1,814 different thumbs. An overview of our modifications to DeepPrint is shown in Figure 5.13.

During the authentication or search stage, the CNN accepts a 1,900 ppi infant fingerprint as input and outputs a 192-dimensional fixed-length representation of the fingerprint. This representation can be compared to previously enrolled representations via the cosine distance between two given representations at 10 million comparisons/second on an Intel i9 processor with 64 GB of RAM. More formally, given an enrollment representation $\mathbf{e} \in \mathbb{R}^{192}$ and a probe representation $\mathbf{p} \in \mathbb{R}^{192}$, a texture matching score s_t is computed as the inner product between $\mathbf{e}$ and $\mathbf{p}$:

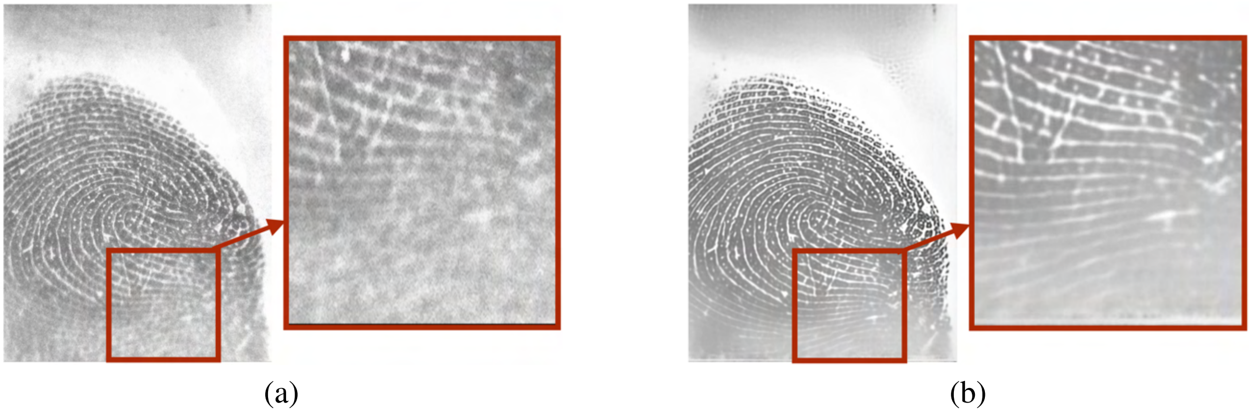
$$s_t = \mathbf{e}^{\mathbf{T}}\mathbf{p} \tag{5.5.3}$$

Note, in our preliminary study [82], we also used a deep learning based texture matcher similar to DeepPrint, however, we did not incorporate minutiae domain knowledge into the texture matcher as is done in DeepPrint (shown in Fig. 5.13). Adopting the strategy of DeepPrint in incorporating minutiae domain knowledge into the deep network further improves the infant recognition performance. We show this quantitatively in the experimental results.

5.5.4 Latent Fingerprint Matcher

Finally, in addition to a state-of-the-art minutiae matcher (supplemented by our high resolution minutiae extractor) and the fine-tuned texture matcher, we include a state-of-the-art latent fingerprint matcher[13] to the final infant fingerprint recognition algorithm. Before using the latent fingerprint matcher to enroll a template, we first include two preprocessing steps: (i) enhancement, and (ii) aging. These preprocessing steps are further described in the following subsections.

[13] We cannot release the name of the matcher because of a NDA, but it is one of the top performing algorithms in the NIST ELFT evaluation [79].



(a) (b)

Figure 5.14 Infant fingerprint (a) before enhancement and (b) after enhancement. Looking inside the small window (red square) we can see that the enhanced infant fingerprint (b) has noticeably improved sharpness and clarity throughout the friction ridge pattern when compared to (a).

5.5.4.1 Enhancement

Due to the low quality of the infant fingerprints (motion blur, wet, dry), we incorporate an enhancement module to improve the sharpness and clarity of the infant friction ridge pattern. In particular, we incorporate a state-of-the-art image super resolution model, Residual Dense Network (RDN) [190]. To retrain RDN for infant fingerprint enhancement, we first add random noise (random kernel) to the training dataset (9,683 images from [84]), followed by a gaussian blur to simulate various types of noise in the infant fingerprint images. Then, we retrain the RDN network (8x version with a modified stride length) to regress to the clean infant fingerprint images. An example of an infant fingerprint before and after enhancement is shown in Figure 5.14.

5.5.4.2 Image Aging

In a similar manner to the strategy we used to age our extracted minutiae sets, we age the enhanced fingerprint *images* prior to passing them to the latent fingerprint matcher. The COTS latent matcher SDK does not accept a minutiae set and as such, we must directly age the images prior to passing them to the matcher. Therefore, if an infant's fingerprint image is captured at an age of less than 3 months, we resize the image with bicubic interpolation by a scalar factor of $\lambda = 1.1$. The scalar factor is the same as that used to scale our minutiae sets. Finally, after enhancement and image aging, we finish the latent preprocessing by resizing all images by a scalar of 0.5 in order to bring the 1,900 ppi fingerprint images to similar size as the adult fingerprint images the latent matcher is designed to operate on (this same procedure was utilized in [84]).

After preprocessing the infant fingerprint images via enhancement and aging, we can enroll the infant images via the latent SDK, and subsequently compute a match score s_l.

5.5.5 Final Match Score

Our final match score s_f is a fusion of a minutiae matcher, texture matcher, and latent matcher. In particular, given our minutiae matching score of s_m, our texture match score s_t as defined in Equation 5.5.3, and our latent match score s_l, our final match score s_f is computed by first normalizing each score (min-max normalization) to a range of $(0, 1)$ and then performing sum score fusion via:

$$s_f = \lambda_m \cdot s_m + \lambda_t \cdot s_t + \lambda_l \cdot s_l \tag{5.5.4}$$

where λ_m, λ_t, and λ_l are set to $0.6, 0.1, 0.3$ using our validation set of 610 manually marked fingerprint images in conjunction with a grid search.

5.6 Experimental Results

In our experimental results, we first show the authentication and search performance for all the infants in our dataset where enrollment occurs during 0-3 months of age, and authentication or search commences 3 months later. We first focus on a 3 month time lapse for the following reasons. (i) Most of our longitudinal data (121 subjects) has a time lapse of 3 months. (ii) Jain *et al.* already show that once infants reach the age of 6 months, they can be enrolled and recognized a year later. In this work, our primary aim is to bridge the gap between 0-3 months (when first time vaccinations commence) and 6 months. If we can effectively recognize the infants enrolled at 2-3 months and

Table 5.4 Infant Authentication Accuracy[3] ($0-3$ months at enrollment with 3 month time lapse between enrollment and authentication)

Algorithm	Enrollment Age: 0-1 months (17 subjects)	Enrollment Age: 1-2 months (36 subjects)	Enrollment Age: 2-3 months (83 subjects)
DeepPrint [48]	17.6%, 29.4%	27.8%, 58.3	45.8%, 68.7%
Verifinger[1]	41.2%, 58.8%	47.2%, 55.6%	79.5%, 86.7%
Latent Matcher[2]	41.2%, 47.1%	50.0%, 61.1%	84.3%, 91.6%
DeepPrint + Verifinger	52.9%, 64.7%	55.6%, 75.0%	86.7%, 89.2%
DeepPrint + Latent Matcher	41.2%, 58.8%	52.8%, 72.2%	85.5%, 91.6%
Verifinger + Latent Matcher	52.9%, 64.7%	58.3%, 75.0%	91.6%, 92.8%
DeepPrint + Verifinger + Latent Matcher	**64.7%, 70.6%**	**63.9%, 83.3%**	**92.8%, 95.2%**

[1] Minutiae are extracted with our high-resolution minutiae extractor, then aged and fed into the Verifinger v11 ISO Matcher.
[2] Images are enhanced, aged, and then fed into a state-of-the-art COTS Latent Matcher.
[3] Reporting TAR @ FAR=0.1%,1.0%

authenticated or searched at 5-6 months, we can re-enroll the infants and continue to recognize the infants longitudinally as shown in [84].

We conclude the experiments by showing the authentication and search performance of Infant-Prints when the time lapse between the enrollment and probe images is extended to a year.

5.6.1 Experimental Protocol

To boost the infant recognition performance, we fuse scores from both of the infant's thumbs and also across the multiple impressions captured during the enrollment session and authentication or search session. For example, if we successfully captured 2 fingerprint images of each thumb in the enrollment session and authentication session, we would compute a total of 8 scores using Equation 5.5.4. These 8 scores are then fused using average fusion.

We also utilize the gender of the infant to further improve the recognition performance. In particular, if two infants have a different gender, we set the matching score to 0.

Table 5.5 Infant Search Accuracy[3] ($0 - 3$ months at enrollment with 3 month time lapse between enrollment and search)

Algorithm	Enrollment Age: 0-1 months (17 subjects)	Enrollment Age: 1-2 months (36 subjects)	Enrollment Age: 2-3 months (83 subjects)
DeepPrint [48]	52.9%, 58.8%	63.9%, 75.0	90.4%, 92.8%
Verifinger[1]	58.8%, 64.7%	69.4%, 77.8%	90.4%, 91.6%
Latent Matcher[2]	52.9%, 58.8%	63.9%, 75.0%	90.4%, 92.8%
DeepPrint + Verifinger	58.8%, 64.7%	69.4%, 77.8%	90.4%, 91.6%
DeepPrint + Latent Matcher	52.9%, 58.8%	63.9%, 75.0%	90.4%, 92.8%
Verifinger + Latent Matcher	58.8%, 58.8%	72.2%, 80.6%	90.4%, 91.6%
DeepPrint + Verifinger + Latent Matcher	**58.8%, 58.8%**	**72.2%, 77.8%**	**90.4%, 91.6%**

[1] Minutiae are extracted with our high-resolution minutiae extractor, then aged and fed into the Verifinger v11 ISO Matcher.
[2] Images are enhanced, aged, and then fed into a state-of-the-art COTS Latent Matcher.
[3] Reporting Rank 1, Rank 5 search accuracy

All imposter scores are computed by comparing impressions from one subject (both thumbs) in a particular session to impressions from another subject (both thumbs) in another session (making sure to only compare impressions if they belong to the same thumb).

5.6.2 Infant Authentication

Table 5.4 shows the authentication performance of the different matchers (as well as the fused matchers) on infants enrolled between the ages of 0-3 months, and authenticated 3 months later. From these results, we observe that none of the individual matchers perform particularly well on any of the age groups when run standalone. However, after fusing the 3 matchers together, we start to get reliable authentication results when the enrollment age is 2-3 months. While the longitudinal authentication results are not yet robust for the age groups of 0-1 months and 1-2 months, we note that vaccinations commence by the age of 3 months. By obtaining promising authentication results at enrollment ages of less than 3 months, we show that fingerprint authentication of infants is indeed a potential solution for providing infants an identity for life.

5.6.3 Infant Search

Table 5.5 shows the Rank 1 search accuracy of Infant-Prints on infants enrolled between the ages of 0-3 months, and searched 3 months later. The gallery size for our search experiment includes every infant which was enrolled in our study (315 infants). We acknowledge that this gallery size is small, however, we note that (i) obtaining a large gallery of infants would require significant resources, man-hours, and IRB regulations and approvals, and (ii) in several applications, it is very possible that the gallery size would be of similar size to ours. For example, if the clinic which we collected our data at were to use Infant-Prints, they would only need to manage a gallery of 315 infants, since that is the total number of infants visiting the clinic in a 1 year time period.

We note from the results of Table 5.5 that Infant-Prints is able to enroll infants at an age of 2-3 months, and search them 3 months later with a Rank 1 search accuracy of 90.4%. While work remains to be done to further improve the performance to say 99%, we note that this is the first study to show promising longitudinal search performance for infants enrolled at ages as young as 2 months.

It can also be seen from Table 5.5 that each individual matcher is able to obtain the same Rank-1 search performance (for the 2-3 month enrollment group) as the fused matcher. We acknowledge that this can likely be explained by the small gallery size, *i.e.* each individual matcher is sufficient to accurately retrieve the fingerprints from the smaller gallery. Given a larger gallery, it is likely that the fused matcher would be necessary to maintain accurate search performance. Obtaining a large scale infant dataset is an area of future research.

We also highlight that DeepPrint is able to obtain much higher search performance than authentication performance (Table 5.4 vs. Table 5.5). This can be attributed to DeepPrint often times outputting high imposter scores (creating false accepts and reducing the authentication accuracy,

Table 5.6 Ablated Infant Authentication Accuracy[4] ($0-3$ months at enrollment with 3 month time lapse between enrollment and authentication)

Algorithm[†]	Enrollment Age: 0-1 months (17 subjects)	Enrollment Age: 1-2 months (36 subjects)	Enrollment Age: 2-3 months (83 subjects)
w/o High Resolution Minutiae Extractor	35.3%, 70.6%	63.9%, 83.3%	90.4%, 95.2%
w/o Aging and Enhancement	47.1%, 64.7%	50.0%, 72.2%	86.7%, 92.8%
w/o Finetuning DeepPrint	58.8%, 64.7%	58.33%, 69.4%	90.4%, 95.2%
w/o Gender	58.8%, 64.7%	52.8%, 80.6%	89.2%, 94.0%
w/o All[1]	35.3%, 47.1%	44.4%, 66.7%	86.7%, 92.8%
with All[2,3]	**64.7%, 70.6%**	**63.9%, 83.3%**	**92.8%, 95.2%**

[1] Algorithm used in our preliminary study [82].
[2] Minutiae are extracted with our high-resolution minutiae extractor, then aged and fed into the Verifinger v11 ISO Matcher.
[3] Images are enhanced, aged, and then fed into a state-of-the-art COTS Latent Matcher.
[4] Reporting TAR @ FAR=0.1%, FAR=1.0%
[†] Each row removes only the modules mentioned in that row.

whereas in search high imposters are not as problematic as long as the true mate gives the highest score).

5.6.4 Ablations

To highlight the hardware and algorithmic contributions of Infant-Prints, we show an algorithmic ablation study in Tables 5.6, 5.7, 5.8, and 5.9, and a hardware ablation study in Table 5.10.

From Table 5.6, we see the performance of the "fused matcher" (Verifinger + COTS Latent Matcher + DeepPrint) following every algorithmic improvement (high-resolution minutiae extraction, aging, enhancement, finetuning DeepPrint, and gender meta-data). Notably, each algorithmic improvement contributes to the overall best performance shown in the final row. We also note that

Table 5.7 Ablated Verifinger Performance

Algorithm	0-1 months (17 subjects)[2]	1-2 months (36 subjects)	2-3 months (83 subjects)
Verifinger	17.6%[1]	36.1%	74.7%
Verifinger + Aging	23.5%	44.4%	74.7%
Verifinger + Aging + Enhancement	29.4% (35.3%)[4]	52.8% (63.9%)	85.5% (90.4%)
Verifinger + Aging + Enhancement + HR Minutiae[3]	41.2% (64.7%)	47.2% (63.9%)	79.5% (92.8%)

[1] TAR @ FAR = 0.1% after a time lapse of 3 months from enrollment age.

[2] Indicates enrollment ages (authentication occurs 3 months later).

[3] HR Minutiae denotes a minutiae set extracted by our high-resolution, infant minutiae extractor, and fed into Verifinger's matcher.

[4] Performance when fused with other matchers (shown in parenthesis) demonstrates that although HR Minutiae does not help the stand-alone performance of Verifinger, it does help when fusing with the other matchers. This is explained further in the text.

our algorithm (last row of Table 5.6) is significantly improved over our previous algorithm (second to last row of Table 5.6) used in our preliminary study [82].

In Tables 5.7 and 5.8 we note that aging and enhancement both improve the "stand-alone" performance of Verifinger and the COTS latent matcher. Although our high-resolution minutiae extractor does not improve the stand-alone performance of Verifinger ("HR Minutiae" in Table 5.7), it does help when fusing Verifinger with the other matchers (as shown in parenthesis). The reason for this is because the Verifinger minutiae extractor performs worse than our HR minutiae extractor on low quality, noisy fingerprints, but better than our minutiae extractor on higher quality images. By improving Verifinger on the lower quality image pairs with our HR minutiae extractor, we can improve the fused matching performance, since the other matchers are already sufficient to hold the matching performance on the higher quality pairs. This can be seen visually in Figure 5.15. When extracting minutiae with Verifinger (Fig. 5.15) (a)), many spurious minutiae are marked, and

Table 5.8 Ablated COTS Latent Matcher (LM) Performance

Algorithm	0-1 months (17 subjects)[2]	1-2 months (36 subjects)	2-3 months (83 subjects)
COTS LM[3]	35.3%[1]	41.7%	77.1%
COTS LM + Aging	35.3%	44.4%	80.7%
COTS LM + Aging + Enhancement	41.2%	50.0%	84.3%

[1] TAR @ FAR = 0.1% after a time lapse of 3 months from enrollment age.

[2] Indicates enrollment ages (authentication occurs 3 months later).

[3] COTS LM does not enable using our own HR minutiae set.

Table 5.9 Ablated DeepPrint Performance

Algorithm	0-1 months (17 subjects)[2]	1-2 months (36 subjects)	2-3 months (83 subjects)
DeepPrint	11.8%[1]	22.2%	41.0%
DeepPrint + Finetuning	17.6%	27.8%	45.8%

[1] TAR @ FAR = 0.1% after a time lapse of 3 months from enrollment age.

[2] Indicates enrollment ages (authentication occurs 3 months later).

Verifinger is unable to establish any true minutiae correspondences between the enrollment image and the probe image. In contrast, our minutiae extractor extracts the minutiae more reliably on this low quality fingerprint pair (Fig. 5.15) (b)), enabling Verifinger to establish enough minutiae correspondences to flip the example pair from a False Reject to a True Accept.

Table 5.9 shows the ablated performance of DeepPrint. Finetuning the model on infant fingerprints again boosts the performance. Although the performance of DeepPrint is lower than the other matchers stand-alone, it still boosts the overall matching performance (Table 5.4) when fused with other matchers due to the complementary texture features it extracts. We do not age fingerprints prior to DeepPrint extraction since DeepPrint is trained on images of varying scale as a data augmentation method during training. Furthermore, we do not enhance images prior to DeepPrint

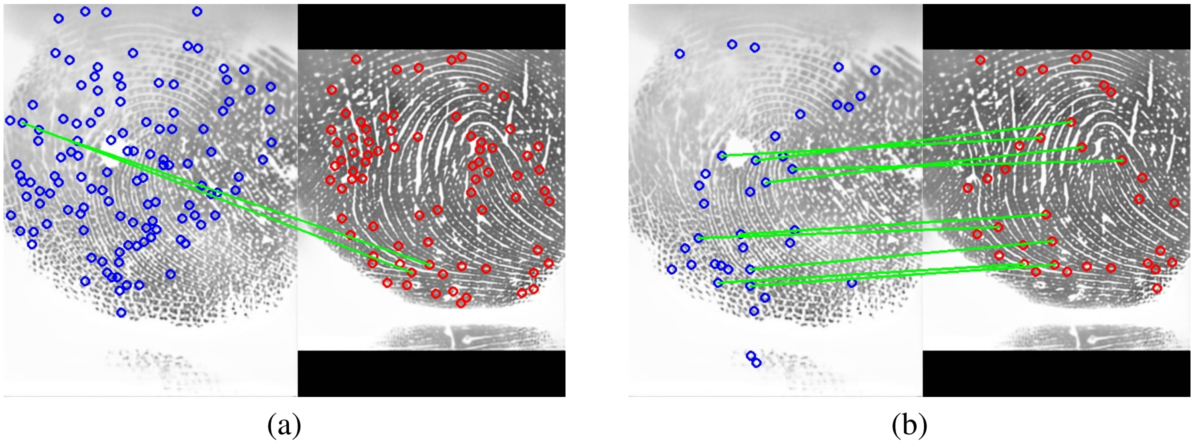


Figure 5.15 Flipping a False Reject case to a True Accept by using our high-resolution minutiae extractor. (a) Minutiae are both extracted and matched using Verifinger. The significant number of spurious minutiae extracted by Verifinger render it impossible for Verifinger to establish minutiae correspondences. (b) Minutiae are extracted using our high-resolution minutiae extractor and subsequently fed into Verifinger. Because our minutiae extractor is much more resistant to spurious minutiae (on infant fingerprints) than Verifinger's minutiae extractor, the Verifinger matcher is able to establish enough true minutiae correpondences to flip this False Reject to a True Accept. Quantitatively speaking, the Verifinger match score is improved from 23 to 48.

extraction as our goal is to have DeepPrint extract complementary textural features which may be discarded post-enhancement.

Finally, we show in our hardware ablation study in Table 5.10 that our contact-based high-resolution (1,900 ppi) fingerprint reader enables higher infant fingerprint authentication performance than a COTS 500 ppi contact-based reader (Digital Personna). We note that there are fewer subjects in Table 5.10 than Table 5.4. This is because Table 5.10 only considers those subjects which were collected on both the MSU RaspiReader and the Digital Persona reader. The difference in subject counts on the MSU RaspiReader and the Digital Persona reader can be attributed to failure to captures on the Digital Persona (often times the ergonomics of the Digital Persona reader (Fig. 5.5 (a)) prevented us from imaging the infant's fingerprints before the infant became too distressed).

We also show in Figure 5.16 that the contact-based RaspiReader genuine and imposter scores are much more separated than the contactless-based RaspiReader (TAR=72.9% vs. TAR=35.6% @ FAR=1.0%). We show score histograms (of single finger comparisons) to compare these two

Table 5.10 Ablated Fingerprint Reader Authentication Results

Reader	0-1 months[2] (12 subjects)	1-2 months (31 subjects)	2-3 months (73 subjects)
Digital Persona (500 ppi)	0%[1]	35.5%	52.1%
MSU RaspiReader (1,900 ppi)	58.3%	64.5%	93.2%[3]

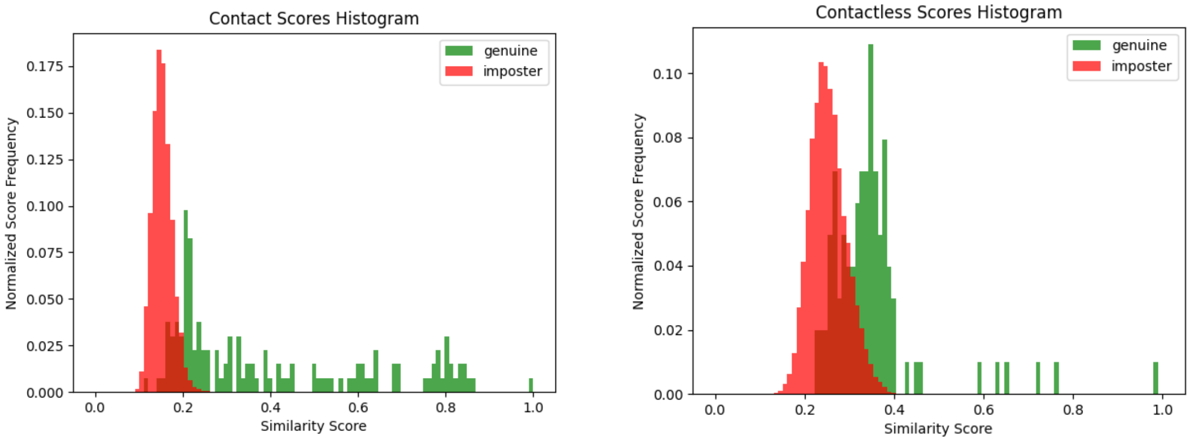
[1] TAR @ FAR = 0.1% after a time lapse of 3 months from enrollment age.

[2] Indicates enrollment ages (authentication occurs 3 months later).

[3] Differs from Table 5.4 because of a different number of subjects.

readers since we only utilized the contactless reader during our last collection session for a limited number of subjects. Our findings of better separation between the contact fingerprint pairs than the contactless fingerprint pairs *contradict* the study of [147] which found that high-resolution, contactless infant fingerprints outperformed high-resolution contact-based infant fingerprints. We found it very difficult to match contactless infant fingerprints since contactless fingerprints have a perspective deformation (certain parts of the finger are further from the camera than others), and the contrast is lower than FTIR fingerprint images. Similar observations about the difficulty of matching contactless fingerprint images have been noted in the literature [103]. In an effort to improve the contactless matching performance, we fine-tuned DeepPrint on $23,416$ contactless fingerprints from $3,276$ fingers from contactless databases released in [36, 103, 110, 148, 192, 193]. We also attempted to normalize the ridge spacing of the contactless fingerprints as was done in [147]. The fine-tuning did improve the contactless matching performance, but did not bridge the gap to the contact fingerprint matching performance.

Example of failure cases (False Accept, False Reject) are shown in Fig. 5.17. These images highlight the difficulty and challenges of doing accurate infant fingerprint recognition over time (moisture, distortion, small inter-ridge spacing, fingerprint aging).



(a) (b)

Figure 5.16 Score Histograms comparing the contact-based RaspiReader with the contactless RaspiReader (single finger performance). Using a contact-based reader shows much better score separation than the contactless reader (TAR=72.9% vs. TAR=35.6% @ FAR=1.0%).

5.6.5 Longitudinal Recognition

Table 5.11 Longitudinal Search Results

Time Lapse: 3 months	Time Lapse: 9 months	Time Lapse: 12 months
95%[1, 2]	90%	90%

[1] Reporting Rank 1 Search Accuracy (Gallery of 315 Infants)
[2] Differs from Table 5.5 because of a different number of subjects.

Table 5.12 Longitudinal Authentication Results

Time Lapse: 3 months	Time Lapse: 9 months	Time Lapse: 12 months
95%[1,2]	90%	85%

[1] Reporting TAR @ FAR = 0.1%
[2] Differs from Table 5.4 because of a different number of subjects.

As a final study, we show the longitudinal search accuracy (Table 5.11) and authentication accuracy (Table 5.12) for infants enrolled at 2-3 months. For this experiment, we selected 20 infants from our total of 315 which were present in all 4 sessions of the data collection and were 2-3 months of age at the first time enrollment (since our earlier studies showed that 2-3 months is the age at which recognition first becomes feasible). Although we have more subjects at individual

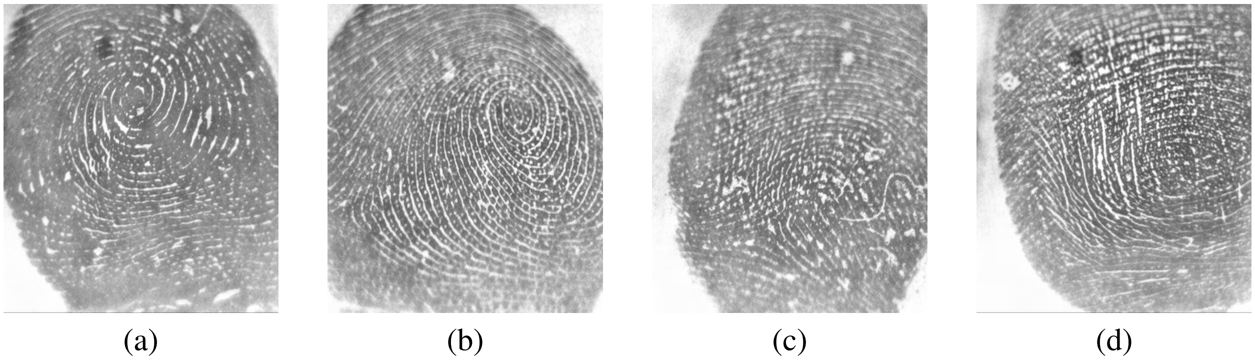


Figure 5.17 Example Infant-Prints failure cases. (a, b) Example of a False Accept due to the similar friction ridge patterns, and the moisture in the enrollment image (a). (c, d) Example of a False Reject due to the motion blur of the uncooperative infant (d). These images highlight several of the challenges in infant fingerprint recognition.

time lapses, we chose the 20 infants which were present in all 4 sessions so that we can observe the impact that time has on the recognition performance whilst fixing the subjects used in the experiments.

Tables 5.11 and 5.12 show that the authentication and search performance stays relatively stable over time. In particular, from 3 months of elapsed time to 9 months of elapsed time, only one infant drops off from being properly searched or authenticated. From 9 months to 12 months, the search accuracy remains unchanged, while only one fewer infant is unable to be authenticated.

Notably, these are the first results to show that it is possible to enroll infants at 2 months old and authenticate them or search them a year later with relatively high accuracy. This highlights the applicability of fingerprints to address the challenges of this chapter. Namely, *can we recognize an infant from their fingerprints in order to better facilitate accurate and fast delivery of vaccinations and nutritional supplements to infants in need.*

5.7 Summary

A plethora of infants around the world continue to suffer and die from vaccine related diseases and malnutrition. A major obstacle standing in the way of delivering the vaccinations and nutrition needed to the infants most in need is the means to quickly and accurately identity or authenticate an infant at the point of care. To address this challenge, we proposed Infant-Prints, and end-to-end infant fingerprint recognition system. We have shown that Infant-Prints is capable of enrolling infants as young as 2 months of age, and recognizing them an entire year later. This is the first ever study to show the feasibility of recognizing infants enrolled this young after this much time gap. It is our hope that this feasibility study and Infant-Prints motivate a strong push in the direction of fingerprint based infant fingerprint recognition systems which can be used to alleviate infant suffering around the world. In doing so, we believe that the work outlined in this chapter will make a major dent in Goal #3 of the United Nations Sustainable Development Goals, namely, *"Ensuring healthy lives and promoting well-being for all,* ***at all ages.****"*

Chapter 6

Summary

6.1 Contributions

In this thesis, we have worked to develop fingerprint recognition systems which are more i) robust, ii) secure, iii) fast, and iv) applicable to all ages. These specific contributions are listed below.

- **Robust:** We have improved the robustness and reliability of fingerprint recognition systems through the design and manufacturing of Universal 3D Wearable Fingerprint Targets. Our Universal Fingerprint Targets can be imaged by all major types of fingerprint sensing technologies (unlike existing 3D fingerprint targets) and are thus useful for fingerprint reader interoperability studies and other operational evaluations of fingerprint readers. By evaluating fingerprint readers with realistic fingerprint targets, as opposed to the existing trivial 2D calibration patterns, fingerprint readers can be better assessed and improved.

- **Fingerprint Reader Security:** We have enhanced the security of fingerprint recognition systems via our open-source, 1,900 ppi RaspiReader. RaspiReader uses a built-in spoof (fake fingerprint attack) detection algorithm based upon its simultaneously captured direct-view and FTIR fingerprint images. We have demonstrated that RaspiReader obtains state-of-the-art levels of fingerprint spoof detection accuracy. We have also demonstrated that RaspiReader generalizes well to spoofs fabricated from materials not seen during training of the RaspiReader spoof detection algorithm. A DIY video showing the assembly process of RaspiReader (from ubiquitous components easily found on Amazon) has been published to YouTube[1], and the 3D parts and capture software for RaspiReader are open-sourced on Github[2].

- **Fingerprint Template Security:** In addition to securing the fingerprint reader module of fingerprint recognition systems via RaspiReader, we also better secure fingerprint templates with our fixed-length fingerprint representation in conjunction with a fully homomorphic encryption matching scheme. In particular, our 192D DeepPrint representations can be matched within the encrypted domain, without loss of accuracy, in 1.25 milliseconds, preventing hackers from stealing the template in the database and also during the matching routine. In contrast, prevailing systems must either i) unencrypt a template prior to matching (leaving them vulnerable to hackers) or ii) sacrifice system accuracy to keep the templates encrypted at all times.

- **Matching Speed:** By using our 192D DeepPrint representations, we not only improve fingerprint template security, we also enable orders of magnitude faster large scale search. In particular, while prevailing fingerprint matching algorithms utilize expensive graph matching algorithms for comparison, DeepPrint representations can be quickly matched with simple distance metrics. Quantitatively, we showed that a state-of-the-art commercial matcher takes 27 seconds to search a fingerprint against a 1.1 million background. In contrast, DeepPrint takes only 300 milliseconds to search, but obtains comparable levels of Rank-1 search accuracy (DeepPrint: 98.80% vs. COTS: 98.85%).

- **Extension to all Ages:** Finally, we concluded the thesis by working to extend fingerprint recognition systems to all ages. In particular, we developed a high-resolution (1,900 ppi) infant fingerprint reader and an accompanying high-resolution infant fingerprint matcher. We then showed in a study of 315 infants, conducted over a time period of 1 year, that

[1] bit.do/RaspiReader

[2] https://github.com/engelsjo/RaspiReader

we could enroll infants at an age of 2-months and still recognize them over a year later. We call our infant fingerprint matcher Infant-Prints. Infant-Prints could provide significant global good in alleviating child suffering and death around the world via better vaccination tracking and government benefits and assistance.

6.2 Suggestions for Future Work

The following are directions of ongoing and future research:

- **Encrypted Fingerprint Search:** Although DeepPrint enables practical encrypted authentication (1.25 milliseconds per encrypted match), the time for encrypted search against large galleries remains impractical. We are working on developing an encrypted search algorithm that enables fingerprint search in the encrypted domain in real-time [46].

- **Large Scale Fingerprint Synthesis:** A primary benefit of DeepPrint is its ability to do fast large-scale search. However, the largest database available to us to evaluate this search is only 1.1 million fingerprints. One possibility to further evaluate DeepPrint is to synthesize a gallery of 1 billion fingerprints [118]. These fingerprints must be both realistic and unique. Given the increased concern of privacy over publicly available fingerprint data, synthesizing realistic fingerprint data could also be useful for augmenting the training data of DeepPrint.

- **One-class Spoof Detection:** In [45] we developed a one-class classifier for fingerprint spoof detection (using images from RaspiReader) which better generalized to unseen materials. This method can be further developed and combined with existing state-of-the-art two-class fingerprint spoof detection algorithms to obtain state-of-the-art in both the seen and unseen material evaluations.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Aditya Abhyankar and Stephanie Schuckers. Integrating a wavelet based perspiration liveness check with fingerprint recognition. *Pattern Recogn.*, 42(3):452–464, March 2009.

[2] AEP Technology. *NANOMAP-500LS*. http://www.aeptechnology.com/pdfs/NanoMap-500LS.pdf.

[3] Multi camera adapter module for raspberry pi. https://www.arducam.com/multi-camera-adapter-module-raspberry-pi. Accessed: 2017-4-15.

[4] S. S. Arora, A. K. Jain, and N. G. Paulter. 3d whole hand targets: Evaluating slap and contactless fingerprint readers. In *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–8, Sept 2016.

[5] Sunpreet S. Arora, Kai Cao, Anil K. Jain, and Nicholas G. Paulter. Design and fabrication of 3d fingerprint targets. *IEEE Transactions on Information Forensics and Security*, 11(10):2284–2297, October 2016.

[6] Sunpreet S. Arora, Anil K. Jain, and Nicholas G. Paulter. Gold fingers: 3d targets for evaluating capacitive readers. *IEEE Transactions on Information Forensics and Security*, pages 1–1, April 2017.

[7] W Babler. Embryologic development of epidermal ridges and their configurations. *Birth defects original article series*, 27(2):95–112, 1991.

[8] Denis Baldisserra, Annalisa Franco, Dario Maio, and Davide Maltoni. Fake fingerprint detection by odor analysis. In *International Conference on Biometrics*, pages 265–272. Springer, 2006.

[9] Jennelle Baptiste. Resistivity of gold. https://hypertextbook.com/facts/2004/JennelleBaptiste.shtml.

[10] Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Failla, Dario Fiore, Riccardo Lazzeretti, Vincenzo Piuri, Alessandro Piva, et al. A Privacy-Compliant Fingerprint Recognition System based on Homomorphic Encryption and Fingercode Templates. In *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–7. IEEE, 2010.

[11] HC Beck, I Ezon, L Flom, C Pitchford, and L Park. Iris recognition technology in newborns. *Investigative Ophthalmology & Visual Science*, 49(13):2265–2265, 2008.

[12] Bir Bhanu and Xuejun Tan. Fingerprint Indexing Based on Novel Features of Minutiae Triplets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):616–622, 2003.

[13] Rex Black. *Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing*. Wiley Publishing, 3rd edition, 2009.

[14] Vishnu Naresh Boddeti. Secure Face Matching using Fully Homomorphic Encryption. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–10. IEEE, 2018.

[15] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid. Face spoofing detection using colour texture analysis. *IEEE Trans. Information Forensics and Security*, 11(8):1818–1830, 2016.

[16] Julien Bringer and Vincent Despiegel. Binary Feature Vector Fingerprint Representation from Minutiae Vicinities. In *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6. IEEE, 2010.

[17] Kai Cao and Anil K Jain. Latent Orientation Field Estimation via Convolutional Neural Network. In *Biometrics (ICB), 2015 International Conference on*, pages 349–356. IEEE, 2015.

[18] Kai Cao and Anil K Jain. Fingerprint Indexing and Matching: An Integrated Approach. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 437–445. IEEE, 2017.

[19] Kai Cao and Anil K Jain. Automated Latent Fingerprint Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[20] Kai Cao, Dinh-Luan Nguyen, Cori Tymoszek, and Anil K Jain. End-to-end latent fingerprint search. *IEEE Transactions on Information Forensics and Security*, 15:880–894, 2019.

[21] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Fingerprint Indexing based on Minutia Cylinder-Code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):1051–1057, 2010.

[22] Raffaele Cappelli, Matteo Ferrara, and Davide 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.

[23] Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, Jul 1997.

[24] Centers for Disease Control and Prevention. Positive Parenting Tips. https://www.cdc.gov/ncbddd/childdevelopment/positiveparenting/index.html, 2018. [Online; accessed 15-February-2019].

[25] Jae Young Choi, Konstantinos N. Plataniotis, and Yong Man Ro. Using colour local binary pattern features for face recognition. In *Proceedings of the International Conference on Image Processing, ICIP 2010, September 26-29, Hong Kong*, pages 4541–4544, 2010.

[26] Tarang Chugh, Kai Cao, and Anil Jain. Fingerprint spoof detection using minutiae-based local patches. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2017.

[27] Tarang Chugh, Kai Cao, and Anil K Jain. Fingerprint spoof buster: Use of minutiae-centered patches. *IEEE Transactions on Information Forensics and Security*, 13(9):2190–2202, 2018.

[28] Tarang Chugh and Anil K Jain. Fingerprint spoof detector generalization. *IEEE Transactions on Information Forensics and Security*, 16:42–55, 2020.

[29] Paolo Cignoni, Massimiliano Corsini, and Guido Ranzuglia. Meshlab: an open-source 3d mesh processing system. *ERCIM News*, 2008(73):45–46, April 2008.

[30] Catherine C Cooksey, Benjamin K Tsai, and David W Allen. Spectral reflectance variability of skin and attributing factors. In *Proc. SPIE*, volume 9461, page 94611M, 2015.

[31] Harold Cummins and Charles Midlo. *Finger Prints, Palms and Soles: An Introduction to Dermatoglyphics*, volume 319. Dover Publications New York, 1961.

[32] Xiaowei Dai, Jie Liang, Qijun Zhao, and Feng Liu. Fingerprint Segmentation via Convolutional Neural Networks. In *Chinese Conference on Biometric Recognition*, pages 324–333. Springer, 2017.

[33] Luke Nicholas Darlow and Benjamin Rosman. Fingerprint Minutiae Extraction using Deep Learning. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 22–30. IEEE, 2017.

[34] Explainable Artificial Intelligence (XAI). https://www.darpa.mil/program/explainable-artificial-intelligence.

[35] Ashim K Datta. *Advances in Fingerprint Technology*. CRC press, 2001.

[36] Debayan Deb, Tarang Chugh, Joshua Engelsma, Kai Cao, Neeta Nain, Jake Kendall, and Anil K Jain. Matching fingerphotos to slap fingerprint images. *arXiv preprint arXiv:1804.08122*, 2018.

[37] Office of Biometric Identity Management Identification Services. https://www.dhs.gov/obim-biometric-identification-services.

[38] Yaohui Ding and Arun Ross. An ensemble of one-class svms for fingerprint spoof detection across different fabrication materials. In *IEEE International Workshop on Information Forensics and Security, WIFS 2016, Abu Dhabi, December 4-7, 2016*, pages 1–6, 2016.

[39] Bernadette Dorizzi, Raffaele Cappelli, Matteo Ferrara, Dario Maio, Davide Maltoni, Nesma Houmani, Sonia Garcia-Salicetti, and Aurélien Mayoue. Fingerprint and on-line signature verification competitions at icb 2009. In *International Conference on Biometrics*, pages 725–732. Springer, 2009.

[40] Dow Corning. *Sylgard PDMS 184 Data Sheet*. http://www.dowcorning.com/DataFiles/090276fe80190b08.pdf.

[41] Dutch Ministry of the Interior and Kingdom Relations. Evaluation Report Biometrics Trial 2b or not 2b. http://www.dematerialisedid.com/PDFs/88630file.pdf, 2004.

[42] Christopher Edwards and Ronald Marks. Evaluation of biomechanical properties of human skin. *Clinics in Dermatology*, 13(4):375 – 380, 1995. Bioengineering of the Skin.

[43] Joshua J Engelsma, Sunpreet S Arora, Anil K Jain, and Nicholas G Paulter. Universal 3d wearable fingerprint targets: Advancing fingerprint reader evaluations. *IEEE Transactions on Information Forensics and Security*, 13(6):1564–1578, 2018.

[44] Joshua J Engelsma, Kai Cao, and Anil K Jain. Fingerprint match in box. *IEEE Biometrics Theory Applications and Systems*, 2018.

[45] Joshua J Engelsma and Anil K Jain. Generalizing fingerprint spoof detector: Learning a one-class classifier. In *2019 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2019.

[46] Joshua J Engelsma, Anil K Jain, and Vishnu Naresh Boddeti. Hers: Homomorphically encrypted representation search. *arXiv preprint arXiv:2003.12197*, 2020.

[47] Joshua James Engelsma, Kai Cao, and Anil K Jain. Raspireader: Open source fingerprint reader. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[48] Joshua James Engelsma, Kai Cao, and Anil K Jain. Learning a fixed-length fingerprint representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[49] Joshua James Engelsma, Debayan Deb, Kai Cao, Anjoo Bhatnagar, Prem Sewak Sudhish, and Anil K Jain. Infant-id: Fingerprints for global good. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[50] Jude Ezeobiejesi and Bir Bhanu. Latent Fingerprint Image Segmentation using Deep Neural Network. In *Deep Learning for Biometrics*, pages 83–107. Springer, 2017.

[51] TJC Faes, HA Van der Meij, JC De Munck, and RM Heethaar. The electric resistivity of human tissues (100 hz-10 mhz): a meta-analysis of review studies. *Physiological Measurement*, 20(4):R1, 1999.

[52] Vincent Falanga and Brian Bucalo. Use of a durometer to assess skin hardness. *Journal of the American Academy of Dermatology*, 29(1):47 – 51, 1993.

[53] Junfeng Fan and Frederik Vercauteren. Somewhat Practical Fully Homomorphic Encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

[54] Faisal Farooq, Ruud M Bolle, Tsai-Yang Jea, and Nalini Ratha. Anonymous and Revocable Fingerprint Recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2007.

[55] IAFIS FAQs. https://www.fbibiospecs.cjis.gov/Certifications/FAQ.

[56] Next Generation Identification (NGI). https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics/ngi.

[57] NGI Fact Sheet. https://www.fbi.gov/file-repository/ngi-monthly-fact-sheet/view.

[58] Speedmixer. http://speedmixer.com/.

[59] F Galton. Finger prints of young children. *British Association for the Advancement of Science*, 69:868–869, 1899.

[60] Francis Galton. *Finger prints*. Macmillan and Company, 1892.

[61] Gamry electro-chemical spectrometer. https://www.gamry.com/potentiostats/.

[62] Hand anthropometry. https://www.scribd.com/document/361450670/Hand-Anthropometry.

[63] Ed German. The History of Fingerprints. https://onin.com/fp/fphistory.html.

[64] Luca Ghiani, Paolo Denti, and Gian Luca Marcialis. Experimental results on fingerprint liveness detection. In *Proceedings of the 7th International Conference on Articulated Motion and Deformable Objects*, AMDO'12, pages 210–218. Springer-Verlag, 2012.

[65] Luca Ghiani, Abdenour Hadid, Gian Luca Marcialis, and Fabio Roli. Fingerprint liveness detection using binarized statistical image features. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–6. IEEE, 2013.

[66] Luca Ghiani, Gian Luca Marcialis, and Fabio Roli. Fingerprint liveness detection by local phase quantization. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 537–540. IEEE, 2012.

[67] Goodix live finger detection. https://findbiometrics.com/goodix-zte-biometric-sensor-3103187/.

[68] Government of India. AADHAR. Unique Identification Authority of India (UIDAI). https://uidai.gov.in, 2019. [Online; accessed 14-February-2019].

[69] Diego Gragnaniello, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Fingerprint liveness detection based on weber local image descriptor. In *Biometric Measurements and Systems for Security and Medical Applications (BIOMS), 2013 IEEE Workshop on*, pages 46–50. IEEE, 2013.

[70] Diego Gragnaniello, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Local contrast phase descriptor for fingerprint liveness detection. *Pattern Recognition*, 48(4):1050–1058, 2015.

[71] Steven A Grosz, Tarang Chugh, and Anil K Jain. Fingerprint presentation attack detection: A sensor and material agnostic approach. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2020.

[72] Steven A Grosz, Joshua J Engelsma, Nicholas G Paulter, and Anil 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, 2019.

[73] Lumidigm V- Series. https://www.hidglobal.com/products/biometrics/lumidigm/lumidigm-v-series-fingerprint-sensors.

[74] Lin Hong, Yifei Wan, and Anil Jain. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):777–789, August 1998.

[75] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[76] IARPA ODIN program. https://www.iarpa.gov/index.php/research-programs/odin/odin-baa.

[77] Index Mundi. World Age Structure. https://www.indexmundi.com/world/age_structure.html, 2018. [Online; accessed 15-February-2019].

[78] Index Mundi. World Birth Rate. https://www.indexmundi.com/world/birth_rate.html, 2018. [Online; accessed 15-February-2019].

[79] Michael D Indovina, RA Hicklin, and G I Kiebuzinski. Nist evaluation of latent fingerprint technologies: Extended feature sets [evaluation# 1]. Technical report, NIST, 2011.

[80] Fingerprint matching software. https://www.innovatrics.com/idkit-fingerprint-sdk/.

[81] 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.

[82] Joshua J Engelsma, Debayan Deb, Anil Jain, Anjoo Bhatnagar, and Prem Sewak Sudhish. Infant-prints: Fingerprints for reducing infant mortality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 67–74, 2019.

[83] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[84] Anil K Jain, Sunpreet S Arora, Kai Cao, Lacey Best-Rowden, and Anjoo Bhatnagar. Fingerprint recognition of young children. *IEEE Transactions on Information Forensics and Security*, 12(7):1501–1514, 2017.

[85] Anil K Jain, Kai Cao, and Sunpreet S Arora. Recognizing infants and toddlers using fingerprints: Increasing the vaccination coverage. In *IEEE International Joint Conference on Biometrics*, pages 1–8. IEEE, 2014.

[86] Anil K Jain, Yi Chen, and Meltem Demirkus. Pores and ridges: High-resolution fingerprint matching using level 3 features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):15–27, 2006.

[87] Anil K Jain, Karthik Nandakumar, and Abhishek Nagar. Biometric Template Security. *EURASIP Journal on Advances in Signal Processing*, 2008:113, 2008.

[88] Anil K Jain, Karthik Nandakumar, and Arun Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern recognition letters*, 79:80–105, 2016.

[89] Anil K Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti. Fingercode: A Filterbank for Fingerprint Representation and Matching. In *Computer Vision and Pattern Recognition, 1999.*, volume 2, pages 187–193. IEEE, 1999.

[90] Anil K Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti. Filterbank-based Fingerprint Matching. *IEEE Transactions on Image Processing*, 9(5):846–859, 2000.

[91] Anil K. Jain, Salil Prabhakar, and Arun Ross. Fingerprint matching: Data acquisition and performance evaluation. Technical Report MSUTR99-14, Department of Computer Science, Michigan State University, East Lansing, Michigan, March 1999.

[92] Anil K Jain, Arun A Ross, and Karthik Nandakumar. *Introduction to biometrics*. Springer Science & Business Media, 2011.

[93] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2010.

[94] Xiaofei Jia, Xin Yang, Yali Zang, Ning Zhang, and Jie Tian. A cross-device matching fingerprint database from multi-type sensors. In *21st International Conference on Pattern Recognition (ICPR), 2012*, pages 3001–3004. IEEE, 2012.

[95] Xudong Jiang, Manhua Liu, and Alex C Kot. Fingerprint Retrieval for Identification. *IEEE Transactions on Information Forensics and Security*, 1(4):532–542, 2006.

[96] Zhe Jin, Meng-Hui Lim, Andrew Beng Jin Teoh, Bok-Min Goi, and Yong Haur Tay. Generating Fixed-length Representation from Minutiae using Kernel Methods for Fingerprint Authentication. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(10):1415–1428, 2016.

[97] Joint Research Center of the European Commission. Fingerprint Recognition for Children. https://bit.ly/2BKuoIj, 2013.

[98] Keyence. *Keyence Optical Digital Microscope VHX-600*. http://www1.keyence.eu/products/microscope/microscope/vhx6002/vhx6002specifications1.php.

[99] Johannes Kotzerke, Stephen A Davis, Jodie McVernon, and Kathy J Horadam. Steps to solving the infant biometric problem with ridge-based biometrics. *IET Biometrics*, 7(6):567–572, 2018.

[100] Philip Dean Lapsley, Jonathan Alexander Lee, David Ferrin Pare Jr, and Ned Hoffman. Anti-fraud biometric scanner that accurately detects blood flow, April 7 1998. US Patent 5,737,439.

[101] Rubisley P Lemes, Olga RP Bellon, Luciano Silva, and Anil K Jain. Biometric recognition of newborns: Identification using palmprints. In *2011 International Joint Conference on Biometrics (IJCB)*, pages 1–6. IEEE, 2011.

[102] Ruilin Li, Dehua Song, Yuhang Liu, and Jufu 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.

[103] Chenhao Lin and Ajay Kumar. Matching contactless and contact-based conventional fingerprint images for biometrics identification. *IEEE Transactions on Image Processing*, 27(4):2008–2021, 2018.

[104] Eryun Liu. Infant Footprint Recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1653–1660, 2017.

[105] Eryun Liu, Heng Zhao, Jimin Liang, Liaojun Pang, Hongtao Chen, and Jie Tian. Random Local Region Descriptor (RLRD): A New Method for Fixed-length Feature Representation of Fingerprint Image and its Application to Template Protection. *Future Generation Computer Systems*, 28(1):236–243, 2012.

[106] Manhua Liu and Pew-Thian Yap. Invariant Representation of Orientation Fields for Fingerprint Indexing. *Pattern Recognition*, 45(7):2532–2542, 2012.

[107] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim L Wayman, and Anil K Jain. FVC Ongoing. https://biolab.csr.unibo.it/fvcongoing/UI/Form/Home.aspx.

[108] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim L Wayman, and Anil K Jain. FVC2002. http://bias.csr.unibo.it/fvc2002/. 2002.

[109] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim L Wayman, and Anil K Jain. Fvc2004: Third fingerprint verification competition. In *Biometric Authentication*, pages 1–7. Springer, 2004.

[110] Aakarsh Malhotra, Anush Sankaran, Mayank Vatsa, and Richa Singh. On matching finger-selfies using deep scattering networks. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(4):350–362, 2020.

[111] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar. *Handbook of Fingerprint Recognition*. Springer, 2nd edition, 2009.

[112] E. Marasco and C. Sansone. On the robustness of fingerprint liveness detection algorithms against new materials used for spoofing. In *Proc. Int. Conf. Bio-Inspired Syst. Signal Process.*, pages 553–558, 2011.

[113] Emanuela Marasco and Arun Ross. A survey on antispoofing schemes for fingerprint recognition systems. *ACM Comput. Surv.*, 47(2):28:1–28:36, November 2014.

[114] Emanuela Marasco and Carlo Sansone. Combining perspiration-and morphology-based static features for fingerprint liveness detection. *Pattern Recognition Letters*, 33(9):1148–1156, 2012.

[115] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino. Impact of artificial gummy fingers on fingerprint systems. In *Proceedings of SPIE*, volume 4677, pages 275–289, 2002.

[116] Medical Expo, THE ONLINE MEDICAL DEVICE EXHIBITION. http://www.medicalexpo.com/medical-manufacturer/test-phantom-2563.html. Accessed: 2017-2-07.

[117] David Menotti, Giovani Chiachia, Allan da Silva Pinto, William Robson Schwartz, Hélio Pedrini, Alexandre Xavier Falcão, and Anderson Rocha. Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Trans. Information Forensics and Security*, 10(4):864–879, 2015.

[118] Vishesh Mistry, Joshua J Engelsma, and Anil K Jain. Fingerprint synthesis: Search with 100 million prints. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2019.

[119] Shimon K. Modi, Stephen J. Elliott, and Hale Kim. Statistical analysis of fingerprint sensor interoperability performance. In *Proceedings of the 3rd IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 294–299, 2009.

[120] S. Moore. Latest tests of biometrics systems shows wide range of abilities. *IEEE Spectrum Online*, 2004. available at http://spectrum.ieee.org/computing/embedded-systems/latest-tests-of-biometrics-systems-shows-wide-range-of-abilities.

[121] Valerio Mura, Luca Ghiani, Gian Luca Marcialis, Fabio Roli, David A. Yambay, and Stephanie A. C. Schuckers. Livdet 2015 fingerprint liveness detection competition 2015. In *BTAS*, pages 1–6. IEEE, 2015.

[122] Abhishek Nagar, Shantanu Rane, and Anthony Vetro. Privacy and Security of Features Extracted from Minutiae Aggregates. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1826–1829. IEEE, 2010.

[123] Karthik Nandakumar. A Fingerprint Cryptosystem Based on Minutiae Phase Spectrum. In *2010 IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2010.

[124] Dinh-Luan Nguyen, Kai Cao, and Anil K Jain. Automatic Latent Fingerprint Segmentation. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9. IEEE, 2018.

[125] Dinh-Luan Nguyen, Kai Cao, and Anil K Jain. Robust Minutiae Extractor: Integrating Deep Networks and Fingerprint Domain Knowledge. In *2018 International Conference on Biometrics (ICB)*, pages 9–16. IEEE, 2018.

[126] Van Huan Nguyen, Jinsong Liu, Thi Hai Binh Nguyen, Hakil Kim, et al. Universal fingerprint minutiae extractor using convolutional neural networks. *IET Biometrics*, 9(2):47–57, 2019.

[127] Shankar Bhausaheb Nikam and Suneeta Agarwal. Local binary pattern and wavelet based spoof fingerprint detection. *Int. J. Biometrics*, 1(2):141–159, August 2008.

[128] NIST special database 14. https://www.nist.gov/srd/nist-special-database-14.

[129] NIST special database 4. https://www.nist.gov/srd/nist-special-database-4.

[130] Kristin Nixon, Valerio Aimale, and Robert Rowe. Spoof detection schemes. *Handbook of Biometrics*, pages 403–423, 2008.

[131] Rodrigo Nogueira, Roberto Lotufo, and Rubens Machado. Fingerprint liveness detection using convolutional neural networks. *IEEE Trans. Information Forensics and Security*, 11(6):1206–1213, 2016.

[132] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, July 2002.

[133] Michio Okajima. Development of dermal ridges in the fetus. *Journal of Medical Genetics*, 12(3):243–250, 1975.

[134] S. Orandi, F. Byers, S. Harvey, M.D. Garris, S.S. Wood, J.M. Libert, and J.C. Wu. Standard calibration target for contactless fingerprint scanners, NIST patent, May 24 2016. US9349033B2.

[135] Pacific Northwest X-Ray Inc. http://www.pnwx.com/Accessories/Phantoms/Radiology/?Sr=Go&gclid=CPesworu59ECFcS6wAodoZ4MTQ. Accessed: 2017-2-07.

[136] Sharath Pankanti, Salil Prabhakar, and Anil K Jain. On the individuality of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1010–1025, 2002.

[137] Perkin elmer lambda 900 uv-vis spectrometer. http://www.perkinelmer.com/category/uv-vis-spectroscopy-uv.

[138] Javier Preciozzi, Guillermo Garella, Vanina Camacho, Francesco Franzoni, Luis Di Martino, Guillermo Carbajal, and Alicia Fernandez. Fingerprint biometrics from newborn to adult: A study from a national identity database system. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(1):68–79, 2020.

[139] Precise Biometrics. https://precisebiometrics.com/products/fingerprint-spoof-liveness-detection/sensor-vulnerability-analysis/.

[140] Zhenshen Qu, Junyu Liu, Yang Liu, Qiuyu Guan, Ruikun Li, and Yuxin Zhang. A Novel System for Fingerprint Orientation Estimation. In *Chinese Conference on Image and Graphics Technologies*, pages 281–291. Springer, 2018.

[141] F Rahmun and O Bausinger. Best practice fingerprint enrolment standards european visa information system. https://bit.ly/2VcyyQN, 2010.

[142] Ajita Rattani and Arun Ross. Automatic adaptation of fingerprint liveness detector to new spoof materials. In *2014 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE, 2014.

[143] Ajita Rattani, Walter J Scheirer, and Arun Ross. Open set fingerprint spoof detection across novel fabrication materials. *IEEE Trans. on Information Forensics and Security*, 10(11):2447–2460, 2015.

[144] Arun Ross and Anil Jain. *Biometric Sensor Interoperability: A Case Study in Fingerprints*, pages 134–145. Springer, 2004.

[145] Arun Ross and Rohan Nadgir. A thin-plate spline calibration model for fingerprint sensor interoperability. *IEEE Trans. on Knowl. and Data Eng.*, 20(8):1097–1110, August 2008.

[146] Robert K Rowe. Multispectral imaging biometrics, December 2 2008. US Patent 7,460,696.

[147] Steven Saggese, Yunting Zhao, Tom Kalisky, Courtney Avery, Deborah Forster, Lilia Edith Duarte-Vera, Lucila Alejandra Almada-Salazar, Daniel Perales-Gonzalez, Alexandra Hubenko, Michael Kleeman, et al. Biometric recognition of newborns and infants by non-contact fingerprinting: lessons learned. *Gates Open Research*, 3, 2019.

[148] Anush Sankaran, Aakarsh Malhotra, Apoorva Mittal, Mayank Vatsa, and Richa 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.

[149] JK Schneider. Quantifying the dermatoglyphic growth patterns in children through adolescence. https://www.ncjrs.gov/pdffiles1/nij/grants/232746.pdf, 2010.

[150] Patrick Schuch, Simon-Daniel Schulz, and Christoph Busch. Deep Expectation for Estimation of Fingerprint Orientation Fields. In *Biometrics (IJCB), 2017 IEEE International Joint Conference on*, pages 185–190. IEEE, 2017.

[151] Stephanie AC Schuckers. Spoofing and anti-spoofing measures. *Information Security technical report*, 7(4):56–62, 2002.

[152] Mojtaba Sepasian, Cristinel Mares, and Wamadeva Balachandran. Liveness and spoofing in fingerprint identification: Issues and challenges. In *Proceedings of the 4th WSEAS International Conference on Computer Engineering and Applications*, CEA'10, pages 150–158, 2009.

[153] Akihide Shiratsuki, Emiko Sano, Masahiro Shikai, Toshiro Nakashima, T Takashima, Masato Ohmi, and Masamitsu Haruna. Novel optical fingerprint sensor utilizing optical characteristics of skin tissue under fingerprints. In *Photonic Therapeutics and Diagnostics*, volume 5686, pages 80–88. International Society for Optics and Photonics, 2005.

[154] Silicone Solutions. *SS-27S Technical Data Sheet*. available at http://siliconesolutions.com/media/pdf/SS-27STDS.pdf.

[155] SilkID. Silk20-Reader. http://www.silkid.com/wp-content/uploads/2017/02/Silk20-Reader-Brochure-v0.7.pdf, 2019. [Online; accessed 4-May-2019].

[156] Smooth-On. *Silicone Ease Release 200 Technical Data Sheet*. available at https://www.smooth-on.com/tb/files/er200.pdf.

[157] Smooth-On. *Silicone Silc Pig Technical Data Sheet*. available at https://www.smooth-on.com/tb/files/Silc_Pig_Pigments.pdf.

[158] Smooth-On. *Silicone Thinner Technical Data Sheet*. available at https://www.smooth-on.com/tb/files/Silicone_Thinner.pdf.

[159] Dehua Song and Jufu Feng. Fingerprint Indexing based on Pyramid Deep Convolutional Feature. In *IEEE International Joint Conference on Biometrics (IJCB), 2017*, pages 200–207. IEEE, 2017.

[160] Dehua Song, Yao Tang, and Jufu Feng. Aggregating Minutia-Centred Deep Convolutional Features for Fingerprint Indexing. *Pattern Recognition*, 88:397–408, 2019.

[161] Stratsys. *Digital Material Technical DataSheet*. available at http://global72.stratasys.com/~/media/Main/Files/Material_Spec_Sheets/MSS_PJ_PJMaterialsDataSheet.pdf#_ga=2.193296616.455247396.1494258560-1158297905.1493673102.

[162] Stratsys. *Stratasys Connex Printer Machine Specifications*. http://www.stratasys.com/~/media/Main/Files/Machine_Spec_Sheets/PSS_PJ_Connex3.ashx.

[163] Yijing Su, Jianjiang Feng, and Jie Zhou. Fingerprint Indexing with Pose Constraint. *Pattern Recognition*, 54:1–13, 2016.

[164] Yagiz Sutcu, Shantanu Rane, Jonathan S Yedidia, Stark C Draper, and Anthony Vetro. Feature Extraction for a Slepian-Wolf Biometric System using LDPC Codes. In *2008 IEEE International Symposium on Information Theory*, pages 2297–2301. IEEE, 2008.

[165] Yagiz Sutcu, Husrev T Sencar, and Nasir Memon. A Geometric Transformation to Protect Minutiae-based Fingerprint Templates. In *Biometric Technology for Human Identification IV*, volume 6539, page 65390E. International Society for Optics and Photonics, 2007.

[166] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, Inception-resnet and the Impact of Residual Connections on Learning. In *AAAI*, volume 4, page 12, 2017.

[167] Bozhao Tan, Aaron Lewicke, David Yambay, and Stephanie Schuckers. The effect of environmental conditions and novel spoofing methods on fingerprint anti-spoofing algorithms. *IEEE Information Forensics and Security, WIFS*, 2010.

[168] Yao Tang, Fei Gao, Jufu Feng, and Yuhang Liu. Fingernet: An Unified Deep Network for Fingerprint Minutiae Extraction. In *IEEE International Joint Conference on Biometrics (IJCB), 2017*, pages 108–116. IEEE, 2017.

[169] ThorLabs. https://www.thorlabs.com/thorproduct.cfm?partnumber=PS911.

[170] Mitchell Trauring. Automatic comparison of finger-ridge patterns. *Nature*, 197(4871):938–940, 1963.

[171] Unique Identification Authority of India, dashboard summary. https://portal.uidai.gov.in/uidwebportal/dashboard.do.

[172] Umut Uludag, Sharath Pankanti, and Anil K Jain. Fuzzy Vault for Fingerprints. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 310–319. Springer, 2005.

[173] Unicef. Faces, Fingerprints, and Feet. https://data.unicef.org/resources/biometrics/. [Online; accessed 4-October-2020].

[174] Maneesh Upmanyu, Anoop M Namboodiri, Kannan Srinathan, and CV Jawahar. Efficient Biometric Verification in Encrypted Domain. In *International Conference on Biometrics*, pages 899–908. Springer, 2009.

[175] Maneesh Upmanyu, Anoop M Namboodiri, Kannan Srinathan, and CV Jawahar. Blind Authentication: A Secure Crypto-Biometric Verification Protocol. *IEEE Transactions on Information Forensics and Security*, 5(2):255–268, 2010.

[176] Ton Van der Putte and Jeroen Keuning. Biometrical fingerprint recognition: dont get your fingers burned. In *Smart Card Research and Advanced Applications*, pages 289–303. Springer, 2000.

[177] Dayong Wang, Charles Otto, and Anil K Jain. Face Search at Scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1122–1136, 2016.

[178] C I. Watson, G.P. Fiumara, E. Tabassi, S.L Cheng, P.A. Flanagan, and W.J. Salamon. Fingerprint vendor technology evaluation, nist interagency/internal report 8034: 2015. *NIST.IR*, 2014. available at https://dx.doi.org/10.6028/NIST.IR.8034.

[179] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A Discriminative Feature Learning Approach for Deep Face Recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016.

[180] World Food Programme. WFP demands action after uncovering misuse of food relief intended for hungry people in Yemen. https://www1.wfp.org/news/wfp-demands-action-after-uncovering-misuse-food-relief-intended-hungry-people-yemen, 2018. [Online; accessed 15-February-2019].

[181] World Food Programme Insight. These changes show that WFP loves us. https://insight.wfp.org/these-changes-show-that-wfp-loves-us-247f0c1ebcf, 2018. [Online; accessed 15-February-2019].

[182] World Health Organization. Prevention not cure in tackling health-care fraud. https://www.who.int/bulletin/volumes/89/12/11-021211/en/, 2011. [Online; accessed 15-February-2019].

[183] World Health Organization. Progress and Challenges with Achieving Universal Immunization Coverage. https://www.who.int/immunization/monitoring_surveillance/who-immuniz.pdf, 2018. [Online; accessed 15-February-2019].

[184] Haiyun Xu, Raymond NJ Veldhuis, Asker M Bazen, Tom AM Kevenaar, Ton AHM Akkermans, and Berk Gokberk. Fingerprint Verification using Spectral Minutiae Representations. *IEEE Transactions on Information Forensics and Security*, 4(3):397–409, 2009.

[185] David Yambay, Luca Ghiani, Paolo Denti, Gian Luca Marcialis, Fabio Roli, and Stephanie A. C. Schuckers. Livdet 2011 - fingerprint liveness detection competition 2011. In *Proc. International Conf. Biometrics*, pages 208–215. IEEE, 2012.

[186] Xi Yin and Xiaoming Liu. Multi-task Convolutional Neural Network for Pose-Invariant Face Recognition. *IEEE Transactions on Image Processing*, 27(2):964–975, 2018.

[187] Soweon Yoon, Jianjiang Feng, and Anil K Jain. Altered fingerprints: Analysis and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):451–464, 2012.

[188] Soweon Yoon and Anil K Jain. Longitudinal Study of Fingerprint Recognition. *Proceedings of the National Academy of Sciences*, 112(28):8555–8560, 2015.

[189] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[190] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[191] Baicun Zhou, Congying Han, Yonghong Liu, Tiande Guo, and Jin Qin. Fast minutiae extractor using neural network. *Pattern Recognition*, 103:107273, 2020.

[192] Wei Zhou, Jiankun Hu, Ian Petersen, Song Wang, and Mohammed Bennamoun. Performance evaluation of 2d to 3d fingerprint recognition. In *2013 6th International Congress on Image and Signal Processing (CISP)*, volume 3, pages 1736–1741. IEEE, 2013.

[193] Wei Zhou, Jiankun Hu, Song Wang, Ian Petersen, and Mohammed Bennamoun. Performance evaluation of large 3d fingerprint databases. *Electronics letters*, 50(15):1060–1061, 2014.

[194] Yanming Zhu, Xuefei Yin, Xiuping Jia, and Jiankun Hu. Latent Fingerprint Segmentation based on Convolutional Neural Networks. In *IEEE Workshop on Information Forensics and Security (WIFS), 2017*, pages 1–6. IEEE, 2017.

[195] Yongfang Zhu, Sarat C Dass, and Anil K Jain. Statistical models for assessing the individuality of fingerprints. *IEEE Transactions on Information Forensics and Security*, 2(3):391–401, 2007.