AN ACCURATE, EFFICIENT, AND ROBUST FINGERPRINT PRESENTATION ATTACK DETECTOR

By

Tarang Chugh

A DISSERTATION

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

Computer Science – Doctor of Philosophy

2020

ABSTRACT

AN ACCURATE, EFFICIENT, AND ROBUST FINGERPRINT PRESENTATION ATTACK DETECTOR

By

Tarang Chugh

The individuality and persistence of fingerprints is being leveraged for a plethora of day-to-day automated person recognition applications, ranging from social benefits disbursements and unlocking smartphones to law enforcement and border security. While the primary purpose of a fingerprint recognition system is to ensure reliable and accurate user recognition, the security of the system itself can be jeopardized by the use of fingerprint presentation attacks (PAs). A fingerprint PA is defined[1] as a presentsation, of a spoof (fake), altered, or cadaver finger, to the data capture system (fingerprint reader) intended to interfere with the recording of the true fingerprint sample/identity, and thereby preventing correct user recognition.

In this thesis, we present an automated, accurate, and reliable software-only fingerprint presentation attack detector (PAD), called *Fingerprint Spoof Buster*. Specifically, we propose a deep convolutional neural network (CNN) based approach utilizing local patches centered and aligned using fingerprint minutiae. The proposed PAD achieves state-of-the-art performance on publicly available liveness detection databases (LivDet) and large-scale government controlled tests as part of the IARPA ODIN program[2]. Additionally, we present a graphical user interface that highlights local regions of the fingerprint image as bonafide[3] or PA for visual examination. This offers significant advantage over existing PAD solutions that rely on a single spoof score for the entire fingerprint image.

Deep learning-based solutions are infamously resource intensive (both memory and processing) and require special hardware such as graphical processing units (GPUs). With the goal of real-time inference in low-resource environments, such as smartphones and embedded devices, we propose

[1] ISO standard IEC 30107-1:2016, https://www.iso.org/standard/53227.html

[2] ODIN, "IARPA-BAA-16-04 (Thor)", https://www.iarpa.gov/index.php/research-programs/odin/odin-baa, 2016.

[3] In the literature, the term live fingerprint has been primarily used to refer a bonafide fingerprint juxtaposed to spoof fingerprints. However, in the context of all forms of presentation attacks, bonafide fingerprint is a more appropriate term as some PAs such as altered fingerprints also exhibit characteristics of liveness [107].

a series of optimizations including simplifying the network architecture and quantizing model weights (for byte computations instead of floating point arithmetic). These optimizations enabled us to develop a light-weight version of the PAD, called *Fingerprint Spoof Buster Lite*, as an Android application, which can execute on a commodity smartphone (Samsung Galaxy S8) with a minimal drop in PAD performance (from TDR = 95.7% to 95.3% @ FDR = 0.2%) in under 100ms.

Typically, deep learning-based solutions are considered as "black-box" systems due to the lack of *interpretability* of their decisions. One of the major limitations of the existing PAD solutions is their poor generalization against PA materials not seen during training. While it is observed that some materials are easier to detect (e.g. EcoFlex) compared to others (e.g. Silgum) when left out from training, the underlying reasons are unknown. We present a framework to understand and interpret the generalization (cross-material) performance of the proposed PAD by investigating the material properties and visualizing the bonafide and PA samples in the multidimensional feature space learned by deep networks. Furthermore, we present two different approaches to improve the generalization performance: (i) a style transfer-based wrapper, called *Universal Material Generator* (UMG), and (ii) a dynamic approach utilizing temporal analysis of a sequence of fingerprint image frames. The two proposed approaches are shown to significantly improve the generalization performance evaluated on large databases of bonafide and PA samples.

Lastly, fingerprint readers based on conventional imaging technologies, such as optical, capacitive, and thermal, only image the 2D surface fingerprint making them an easy target for presentation attacks. In contrast, Optical Coherent Tomography (OCT) imaging technology provides rich depth information, including the internal fingerprint, eccrine (sweat) glands, as well as PA instruments (spoofs) placed over finger skin. As a final contribution, we present an automated PAD approach utilizing cross-sectional OCT depth profile scans which is shown to achieve a TDR of 99.73% @ FDR of 0.2% on a database of $3,413$ bonafide and 357 PA OCT scans, fabricated using 8 different PA materials. We also identify the crucial regions in the OCT scans necessary for PA detection.

To my loving parents, sister, and my love

ACKNOWLEDGMENTS

As my Ph.D. approaches its culmination with this dissertation, I would like to acknowledge the roles of several individuals who were instrumental in the successful completion of my Ph.D. research. Foremost, I want to express my deepest gratitude to my advisor, Prof. Anil K. Jain, for his unwavering support and encouragement to strive for excellence. His scientific intuition, rigor, and passion for research has always inspired me to give my best in all my endeavors. His ability to explain complex things through simple examples, systematic investigation of a scientific problem, and attention to detail, are some of the things that I will always look up to for the rest of my life. Apart from being a great scientist, he is an extraordinary human being with a humble nature and a caring heart. I also want to thank my undergraduate advisors, Prof. Mayank Vatsa and Prof. Richa Singh, for believing in me and encouraging me to pursue higher studies.

I would also like to express my sincere gratitude to my Ph.D. committee, Prof. Arun Ross, Prof. Xiaoming Liu, and Prof. Vidyadhar Mandrekar, for evaluating my work and providing valuable comments and suggestions. I am grateful to Dr. Kai Cao, whose willingness to answer my numerous questions with enthusiasm has nourished my intellectual maturity during the initial years of my Ph.D. I would also like to thank Prof. Jiayu Zhou, Elham Tabassi, and Nicholas G. Paulter Jr., for their invaluable guidance that proved monumental towards the success of several studies related to latent fingerprints, altered fingerprints and fingerprint minutiae extractors. I would also like to thank Prof. Philip Eisenlohr, Dr. Aritra Chakraborty, and Geeta Kumari from Dept. of Chemical Engineering and Material Science, for providing their insights on investigating presentation attack material characteristics; a special thanks to Natalia Pajares for her immense help with the experiments.

Every day during my Ph.D. studies has been a great opportunity for learning, thanks to my colleagues in PRIP Lab, CV Lab, and iPRoBe Lab. Our weekly meetings, valuable discussions and feedbacks significantly influenced my approach to research. A big thank you to Sunpreet, Radha, Lacey, Charles, Inci, Keyur, Debayan, Josh, Sixue, Yichun, Steven, Divyansh, Vishesh,

Joel, Yaojie, Amin, Sudipta, Anurag, Shivangi, Renu, Cunjian, and Thomas. A special thanks to Chris Perry for managing integration of our solutions, assisting with assembling hardware, and being a cheerful person.

I would like to express my sincere gratitude to Dr. Srimat Chakradhar and Dr. Yi Yang, for giving me the opportunity to intern at NEC Labs America, Princeton, New Jersey. It was a wonderful industry experience, where I worked on a very important problem of automated tattoo detection and recognition.

Last but not the least, the invaluable role played by my parents and sister is beyond words. I am blessed with a wonderful family and will always be indebted to them for their everlasting support and encouragement. Over the past five years, *mi amor*, Swati, has stood by me through thick and thin, bringing out the best in me. Being far away from family was not easy, but I am grateful to the friendships, I made here in Michigan, who have become my family now; especially, Yashesh, Vikram, Prakash, Sabya, Kanchan, Garima, Kokil, Mayank, Abhinav, Kamla, Aritra, Sap Da, Preetam, Sayali, Rahul, and many more.

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Algorithms

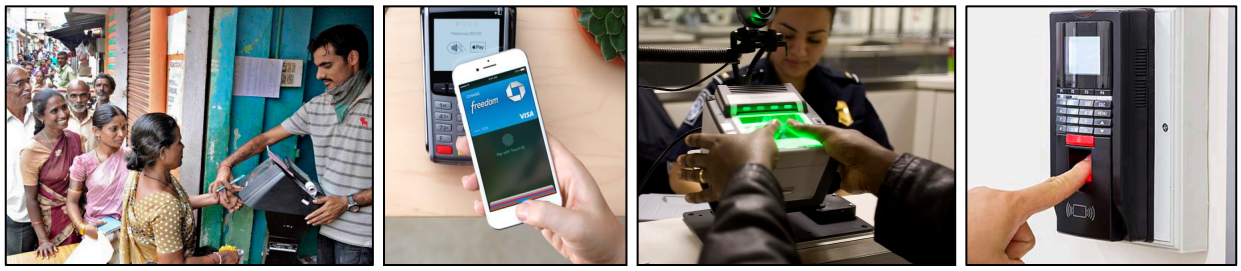
Chapter 1

Introduction

Over 125 years ago, the pioneering work done by Sir Francis Galton brought together and strengthened the evidence essential to the validation of fingerprints as means of personal identification: *permanence* of the fingerprint characteristics, *uniqueness* of ridge details, *variability* and *identifiability* of friction ridge patterns. In his 1892 book titled *"Finger Prints"* [52], he judiciously commented on the potential of friction ridges:

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

Fingerprints have a long history of use as a means of reliably identifying individuals. The earliest recorded use of fingerprints dates back to $1955 - 1913\ BC$, when clay tablets with fingerprints were used to seal business contracts in ancient Babylon. In China, fingerprints were used to sign legal documents by persons without writing skills in $600 - 700\ AD$ [63, 104]. Such historical records indicate an inquisitiveness and perhaps purposeful focus on fingerprints. However,



Public Distribution System | Mobile Payment | International Border Crossing | Access Control

Figure 1.1 Fingerprint recognition based authentication systems used in day-to-day applications. (a) India's Aadhar Program [159], (b) Apple Pay [3], (c) International Border Crossing, US Visit (OBIM) [34], and (d) Access Control [149]. Image Source: Google Images.

the scientific study of fingerprints as a tool of human identification emerged only in the late 19^{th} century [47, 51, 68].

With the advances in science and technology over the last few decades, fingerprint recognition systems have become ubiquitous with its footprint in a plethora of different applications such as mobile payments [3], access control [149], international border crossing [34] and national ID [159] (see Figure 1.1). Although the fingerprint research community has made significant advances over the last few decades, there remains certain challenging avenues in fingerprint recognition where further advances are required.

In this chapter, we first describe the morphology and development process of the friction ridge skin. We then present the fundamental tenets of fingerprints, highlighting two of them which validate its use for personal identification: *uniqueness* and *permanence*. We then discuss the major milestones in the history of fingerprint recognition. Next, we describe the architecture of modern-day automated fingerprint identification systems (AFIS) and discuss the vulnerabilities and research avenues in fingerprint recognition. Finally, we conclude the chapter by presenting the contributions of this dissertation.

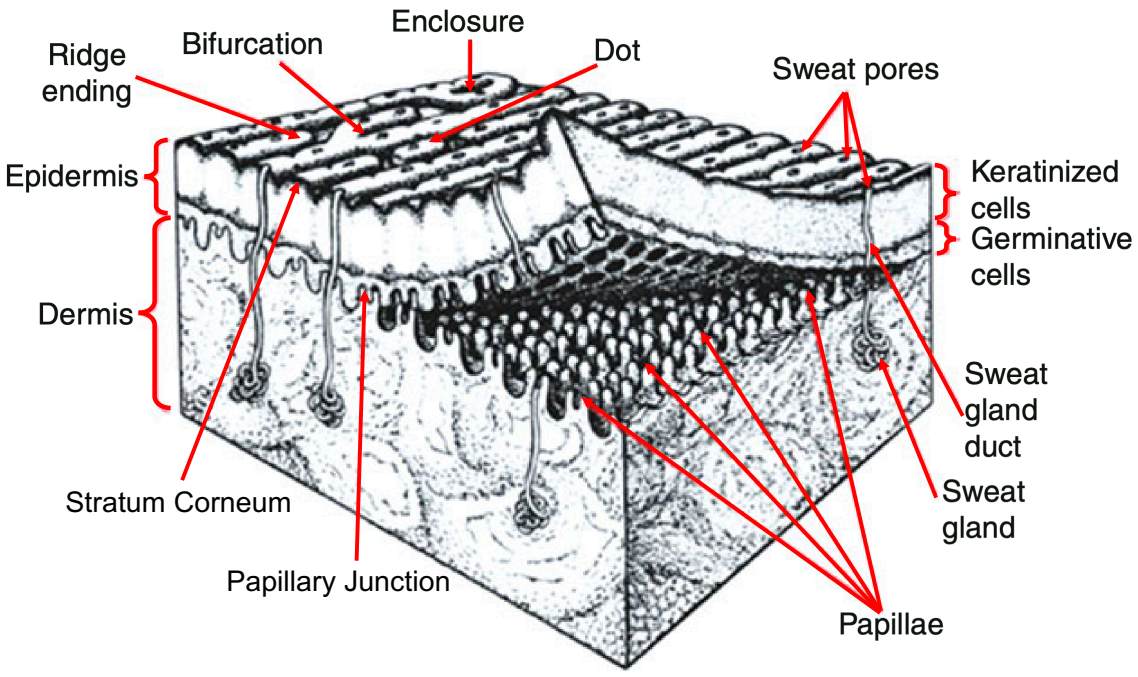


Figure 1.2 Illustration of the morphological structure of the friction ridge skin. Image reproduced from [120].

1.1 Morphology and Development of Friction Ridges

The friction ridge skin is a layered tissue with the outermost layer known as *epidermis* and the external-facing sublayer of epidermis, where the surface fingerprint exists, is known as *stratum corneum* [96]. The layer below epidermis is known as *dermis*, and the junction between epidermis and dermis layers is known as *papillary junction*. There are helically shaped ducts in the epidermis layer connecting the eccrine (sweat) glands in the dermis to the sweat pores on the surface. See Figure 1.2.

Biological evidence suggests that the development of friction ridge begins in late embryological and early fetal development periods and are physiologically present at birth [163]. At $7-8$ weeks of estimated gestational age (EGA), swollen mesenchyme tissue[1] under the epidermis layer on the palmar surface of hands and soles of the feet, called *Volar Pads*, are formed. See Figure 1.3 (a). Subsequently, basal cells[2] of the epidermis layer begin to divide rapidly forming primary ridge

[1]Mesenchyme tissue is a part of the embryo which develops into connective tissue, cartilage, bone, etc. [163]
[2]Basal cells are a type of cell within the skin that produces new skin cells as old ones die off [7].

units which will later become the centers of sweat gland development (Figure 1.3 (b) - (d)) [7]. During $10-11$ weeks of EGA, these ridge units grow and merge into one another along the lines of relief, perpendicular to the compression forces, while forming definitive ridge characteristics, such as ridge bifurcations and endings (Figure 1.3 (e) - (g)). The precise location and orientation of any particular ridge characteristic within the developing ridge field is governed by a random series of infinitely interdependent forces applied across that particular area of skin at that critical moment. These characteristics are believed to be unique because slight differences in the physiological environment, mechanical stress, or variation in the timing of development could significantly affect their location and orientation [163].

During weeks $14-15$ of gestation, the primary friction ridges experience proliferation in two directions: the upward push of new cell growth and the downward penetration of the sweat glands. Typically, the whole volar surface is ridged by 14 weeks of EGA (Figure 1.3 (h)). Between weeks $15-17$ of EGA, sweat pores begin forming and secondary ridges appear between the primary ridges and the underside of the epidermis (Figure 1.3 (i) - (j)). During weeks $17-24$ secondary ridges become completely mature. The secondary ridges (or surface friction ridge pattern) scanned by traditional (optical and capacitive) fingerprint readers are merely an instance or a projection of the primary ridges, a master print existing on the intersection of epidermis and dermis layers (*i.e.*, papillary junction).

During the development of primary friction ridge, the central nervous and cardiovascular systems also undergo a crucial period of development. Disposition of capillary-nerve pairs beneath the dermis layer produces an identical vascular fingerprint with the same individual architecture [141]. These observations suggest the permanence of fingerprints; minor cuts and bruises on the fingers do not change fingerprint patterns because new skin cells are generated beneath the epidermis and facilitate the reformulation of fingerprint patterns on the epidermis.

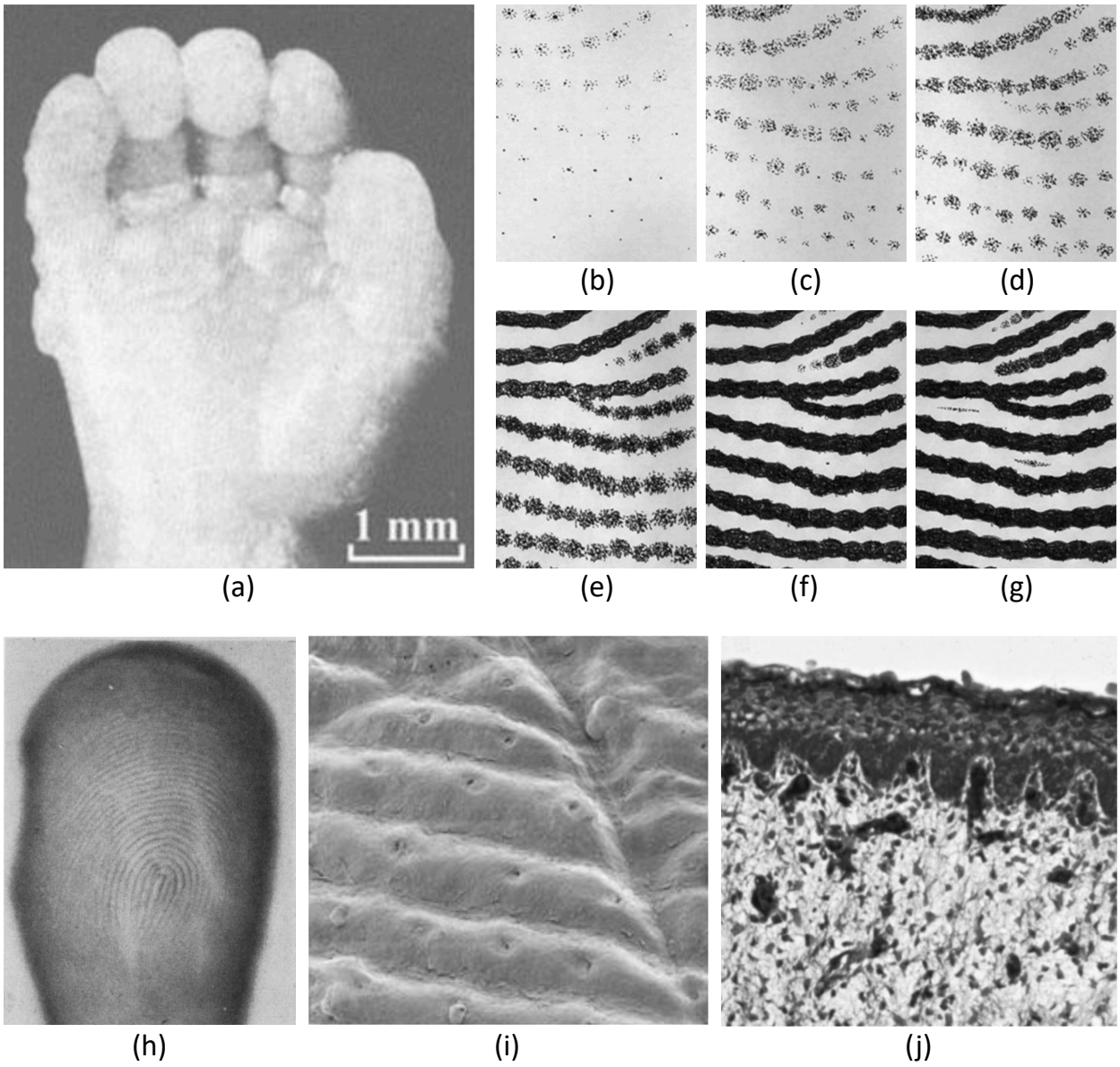


Figure 1.3 Illustration of the fingerprint formation process. (a) Volar pads begin forming during weeks $6-7$ of gestation, (b)-(c) localized ridge units appear, and (d)-(g) ridge units merge to form ridges with unique characteristics during weeks $10-11$, (h) whole volar surface is ridged by 14 weeks, (i) sweat glands and pores begin forming during weeks $14-15$, and (j) secondary ridges begin to form in weeks $15-17$ and are fully matured by 24 weeks of gestation. Images reproduced from [77].

Figure 1.4 Fingerprints of William J. Herschel's son (A. E. H. Herschel) at ages (a) 7, (b) 17, and (c) 40 years. Images reproduced from [69].

1.1.1 Fundamental Tenets of Fingerprint Recognition

In principle, any physiological, behavioral, or anatomical characteristic of an individual can be used as a biometric trait for personal identification. However, there are two fundamental tenets of fingerprints that underlie their wide use for recognizing individuals:

(i) *Uniqueness*: Due to the random forces in play during the formation of friction ridge details, no two fingers, even for the same individual, are identical. Individuals sharing the same DNA, such as monozygatic twins, also have unique fingerprints [84]. Several studies have attempted to assess the individuality of fingerprints [127], however, these studies are either based on relatively simple statistical models of fingerprint characteristics or rely on empirical evaluation involving a small number of subjects.

(ii) *Permanence*: Friction ridge patterns are believed to be persistent during the lifetime of an individual. William Herschel, a German-born British astronomer, was the first to demonstrate the permanence of fingerprints in his 1916 book titled *The Origin of Finger-Printing* [69]. He collected longitudinal inked impressions of his son's finger at the ages of 7, 17, and 40 years old and concluded that fingerprints remained constant over time (see Figure 1.4). In 2015, Yoon and Jain [171] conducted the largest formal study till date involving longitudinal fingerprint records

Figure 1.5 Fingerprints of a subject at ages 34, 40, 42, 43, 44, and 45 years old from the longitudinal database used in [171].

of $15,597$ subjects to assess the permanence of fingerprints (see Figure 1.5). They utilized multi-level statistical models and a state-of-the-art AFIS and concluded that the fingerprint recognition accuracy of the AFIS did not degrade with time (over 12 years for which data is available). This observation asserted that fingerprint recognition accuracy does not change over the lifetime of an individual, despite minor changes in the fingerprint ridge structure due to cuts and bruises.

In addition to uniqueness and permanence, the success of fingerprints as a biometric trait is also attributed to how well it satisfies several key principles: (i) universality, (ii) performance, (iii) user acceptance, (iv) collectability, (v) throughput, (vi) template size, (vii) ease of system integration, and (viii) resistance to spoof and template attacks [86].

1.2 Fingerprint Recognition Milestones

1.2.1 Early Developments

The book *Achaeology in the Holy Land* by Kenyon reports the discovery of thumbprints found in Neolithic bricks from the ancient city of Jericho, State of Palestine, ca. 7000 BC [89]. Similar ancient artifacts with carvings of friction ridge patterns have been found in many places around the world. However, the earliest recorded authentication application of fingerprints dates back to $1955-1913$ BC, when clay tablets with fingerprints were used to seal business contracts in ancient Babylon. In $600-700$ AD China, fingerprints were used to sign contracts and legal documents in the Tang period [63].

1.2.2 Seminal Scientific Studies

While many remnants of fingerprint have been found in history, the scientific study of fingerprints as a tool of human identification emerged only in the 19^{th} century. In 1858, Sir William Herschel as the British chief administrative officer in Bengal, India, mandated use of handprints for civil contracts for payroll distribution to laborers. In 1869 Britain, the Habitual Criminals Act was passed to develop a means to classify the records of habitual criminals (or repeat offenders), such as body measurement, marks, or photograph, to readily re-identify them with certainty [129]. In 1880, Dr. Henry Faulds published a seminal article in Nature suggesting the use of fingerprints for criminal investigations [47]. In 1882, Alphonse Bertillon, a clerk in the Paris Police Identification Bureau, devised a system of recording body measurements (known as *Bertillonage*), which was later adopted throughout France. The first identification using his system was made in February 1883. His anthropometry cards were supplemented with fingerprints on the back side, which led to more identifications compared to any other body measurements [63].

It was the studies by Sir Francis Galton, cousin of Charles Darwin, that brought together and strengthened the evidence essential to the validation of fingerprints as means of personal identification. In 1892, in his seminal book *Finger Prints* [52], he pointed out ridge characteristics which purportedly make each fingerprint unique, such as ridge endings and bifurcations and made the statement that fingerprints remain unchanged throughout the lifetime. In honor of his contributions, the ridge characteristics (now widely known as *minutiae* points) are also called *"Galton"* details.

In 1900, Sir Edward Henry introduced a scientific fingerprint classification system [67], which was later popularly known as *Henry System of Classification*. In 1901, it was officially introduced at New Scotland Yard for criminal identification [63]. In 1963, Mitchell Trauring proposed the first algorithmic approach for comparing friction ridge patterns based on minutiae details [157]. The first Automated Fingerprint Identification Systems (AFIS) became a reality in 1974, avoiding tedious and time consuming manual approach to comparing fingerprints[3].

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

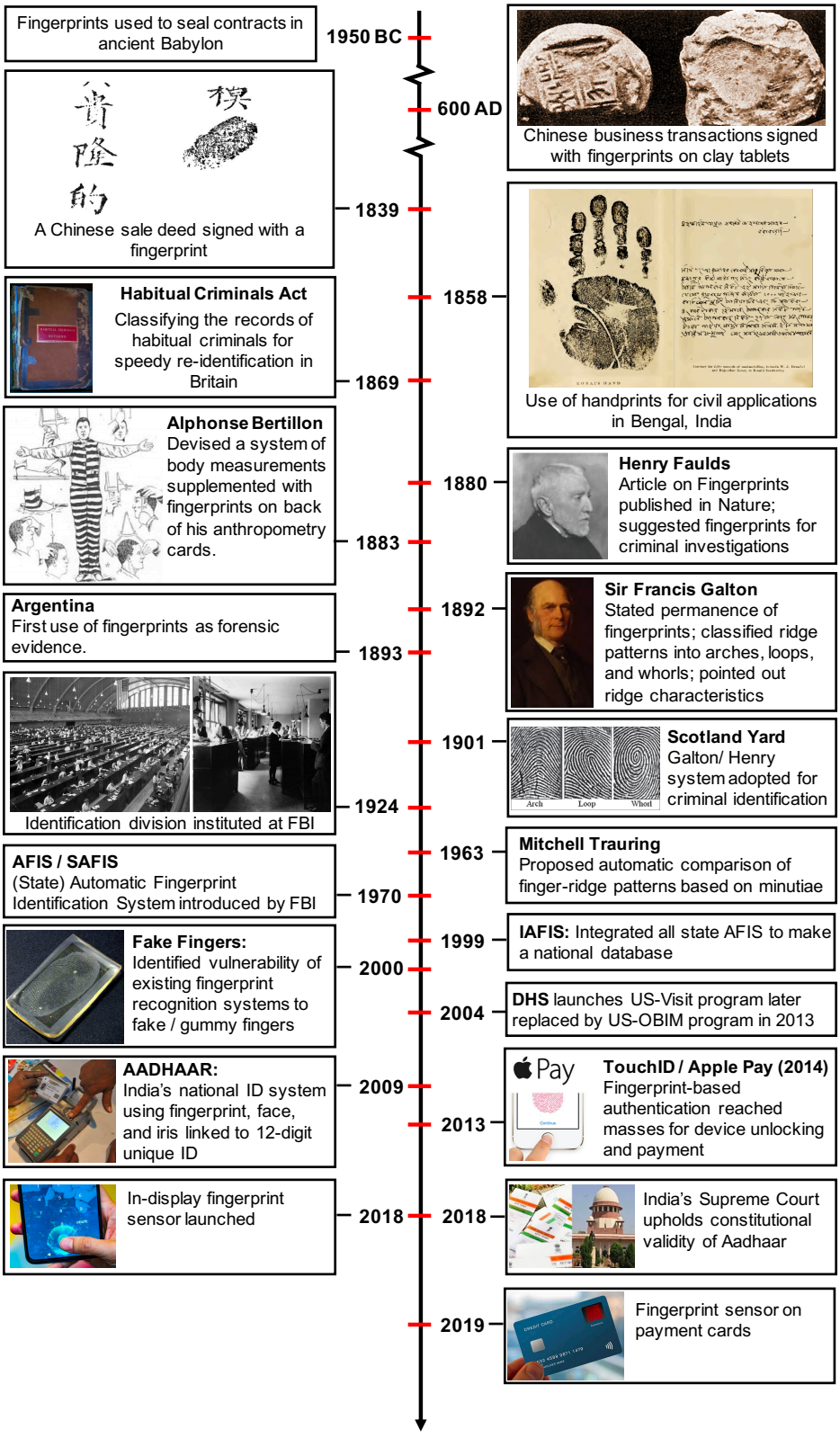


Figure 1.6 Timeline illustrating some of the major milestones in the history of fingerprint recognition.

1.2.3 Landmarks in Law Enforcement Applications

In 1880, Dr. Henry Faulds suggested the use of fingerprints not only for identification, but also for criminal investigations [47]. Thirteen years later in 1893, fingerprints were used for the first time to solve a murder case of two children in Argentina [63]. In 1897 in Bengal, India, another murder case was solved using two brown smudges of fingerprints found on an almanac. Sir Edward Henry, Herschel's successor in India, found the prints to match with an ex-convict Kangali Charan, whose thumbprint was already in the records due to a prior theft conviction [63].

In 1901, use of fingerprints was officially introduced at New Scotland Yard by Sir Edward Henry for criminal identification, replacing the relatively inaccurate Bertillon system. The first fingerprint-based large-scale systematic method of identification was adopted in United States of America in 1902. Dr. Henry Forest installed the new system to inhibit applicants from cheating the New York Civil Service Commission [63]. In the following years, fingerprint-based authentication was adopted in the New York State Prison (1903) and the U.S. Army (1906). Subsequently, a young woman named Mary Holland, studying the Henry system, went throughout the United States teaching the classification system to various law enforcement agencies.

A major development happened in the year 1924, when the United States Congress mandated the collection of fingerprints of criminals. Consequently, a new identification division was instituted at the Federal Bureau of Investigation (FBI). In 1933, a unit specializing in technical analysis of latent fingerprints, *i.e.*, noisy finger marks unintentionally left at a crime scene, was also established at the FBI [120]. With the increasing load to maintain a large repository and perform manual classification of fingerprints, there was a need to automate the process.

A report compiled by the RAND Corporation [62] highlighted the opportunities for much more effective use of physical evidence such as fingerprints, to improve crime solving performance. Recognizing the potential of emerging technology together with electronics revolution happening in 1970s, agencies including the FBI, the UK Home Office, and the Japanese and French police departments undertook research initiatives that led to development of Automated Fingerprint Identification Systems (AFIS) [92]. Law enforcement agencies at the state and local level also began

installing such systems known as State AFIS (SAFIS). Specifically, in 1984, a state AFIS supplied to the authorities in San Fransisco, with a completely new "crime scene to courtroom" philosophy, proved its worth in the real world[4].

In 1999, the FBI launched an Integrated AFIS (IAFIS) which allowed electronic record submission from state and local authorities to the national database and supported capabilities to perform direct large-scale searches in the national repository [92]. It also supported automated tenprint and latent fingerprint searches, electronic exchanges of fingerprints and responses, and text-based searches based on descriptive information. In 2011, IAFIS was upgraded to the Next Generation Identification (NGI) system, with the largest collection of criminal records and enhanced fingerprint recognition capabilities improving fingerprint matching accuracy from 92% to 99.6% with faster response times[5]. It is maintained by the FBI Criminal Justice Information Service (CJIS) and contains fingerprints of more than 145.7 million criminal and civil individuals as of June 2019[6].

1.2.4 Notable Use in Civil and Commercial Applications

In addition to long standing fingerprint applications in law enforcement and forensics, a number of civilian applications are utilizing the individualization property of fingerprints. This has been possible due to the availability of low-cost fingerprint acquisition devices, efficient and robust fingerprint recognition algorithms, and increase in processing power and memory capacity at low prices. For example, a solid state fingerprint reader with fingerprint matching algorithm in a mobile phone costs under US $2 per device.

- **National ID**: In 2009, the Unique Identification Authority of India (UIDAI) launched a national ID system known as *Aadhaar*[7] for the residents of India. Any individual, irrespective of age and gender, can submit their demographic and biometric information (ten fingerprints, two iris, and face photograph) to enroll in the system and obtain a 12-digit unique ID. It is

[4]https://www.gemalto.com/govt/biometrics/afis-history
[5]https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics/ngi
[6]https://www.fbi.gov/file-repository/ngi-monthly-fact-sheet
[7]https://uidai.gov.in/what-is-aadhaar.html

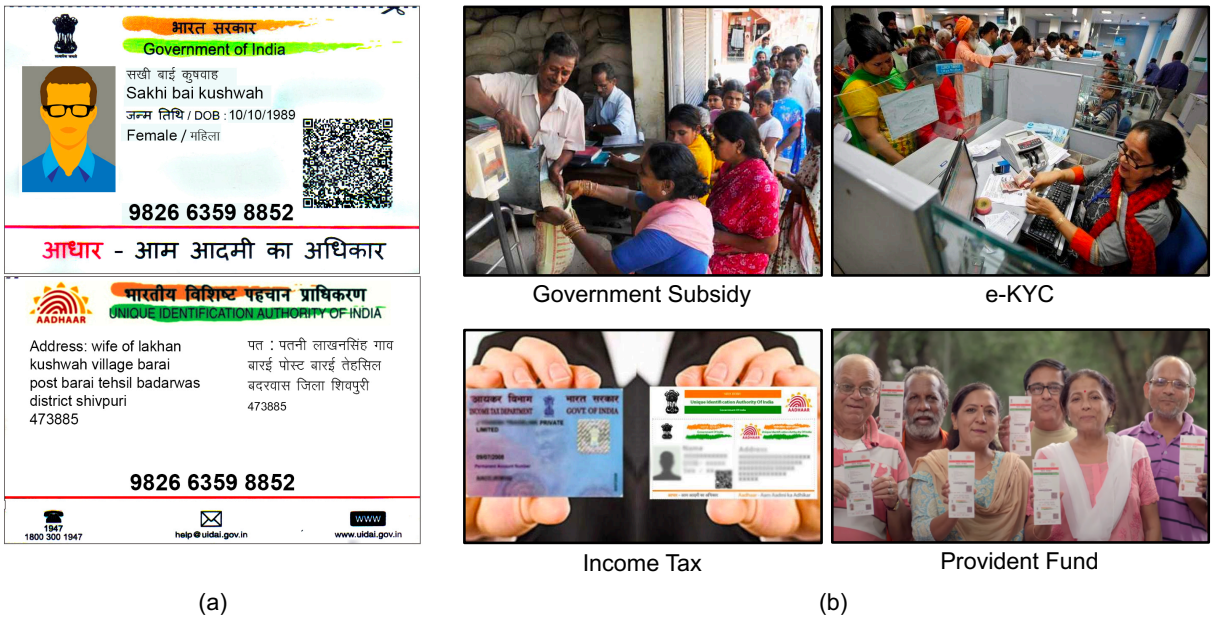
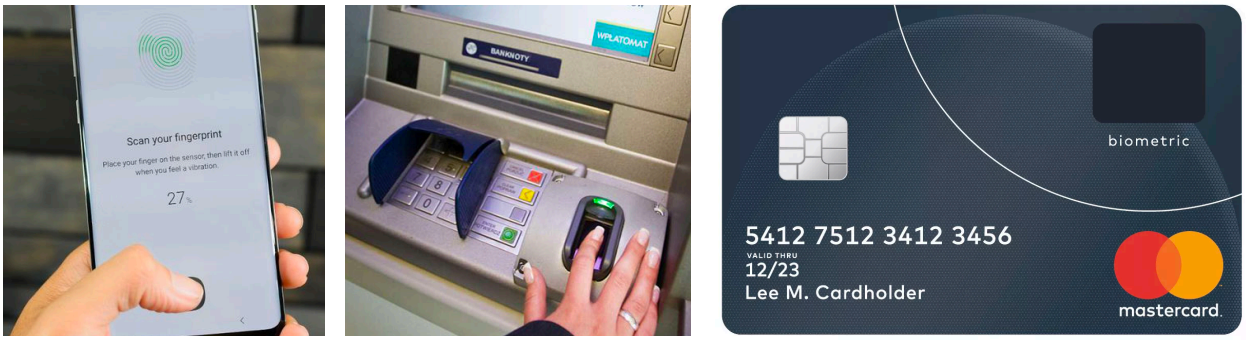


Figure 1.7 India's Aadhaar is the largest biometrics based identification system in the world, with more than 1.25 billion enrollments [159] (March, 2020). (a) A sample Aadhaar ID card containing a 12-digit unique number which is linked to an individual's demographic and biometric information. (b) Some of the applications which utilize Aadhaar ID includes electronic-Know Your Client (e-KYC) service, distribution of government subsidies, processing income tax and employee provident funds.

designed as a strategic policy tool for social and financial inclusion, corruption-free delivery of public sector reforms, managing fiscal budgets, increasing convenience and promoting hassle-free people-centric governance (see Figure 1.7). Biometric information allows the authorities to perform de-duplication at enrollment and online authentication in the field to prevent any misuse. It is by far the largest biometrics based identification system in the world, with more than 1.25 billion enrollments [159] (March, 2020).

- **Infant Fingerprinting**: As of December 2019, there are over 677 million children worldwide in the age group of $0 - 4$ years old[8] and over $370,000$ are born every day[9]. Given that a majority of these childbirths occur in developing countries, where the infant[10] mortal-

[8]UN Data Project: https://bit.ly/2MF9FNs
[9]https://www.indexmundi.com/world/birth_rate.html
[10]The term "infant" is typically applied to young children under one year of age.



(a) (b) (c)

Figure 1.8 Fingerprint-based authentication is used in many commercial applications, including executing financial transactions, unlocking devices, access control, etc. (a) A user enrolling their fingerprint in Samsung Galaxy S10 with an in-display ultrasound-based fingerprint sensor, (b) user authentication in ATM transactions, and (c) biometric-enabled payment cards with embedded fingerprint sensor and on-card storage for fingerprint template.

ity rate can be as high as 180 deaths per 1000 live births[11], fingerprint based identification can provide a form of identity for health care applications such as tracking vaccination and improving nourishment [78]. A low-cost fingerprint reader specifically designed to capture infant fingerprints in the field is shown to achieve an identification accuracy of TAR = 90% @ FAR = 0.1% [45].

- **Commercial Applications**: Due to the rising concerns about data security and financial fraud, coupled with the advent of compact and inexpensive sensors, many commercial organizations have initiated their own deployment of fingerprint-based consumer authentication, especially for access control and secure financial transactions. Many consumer devices, such as laptops and smartphones, utilize solid-state fingerprint readers for device unlocking and making online purchases[12]. In 2018, the global penetration of smartphones with fingerprint sensors reached 67% compared to only 19% in 2014[13]. Mastercard[14] and Visa[15] are conducting pilot programs of utilizing biometric payment cards with embedded fingerprint sensors,

[11] https://www.infoplease.com/world/health-and-social-statistics/infant-mortality-rates-countries
[12] https://support.apple.com/en-us/HT207054
[13] https://www.statista.com/statistics/522058/global-smartphone-fingerprint-penetration/
[14] https://www.mastercard.us/en-us/merchants/safety-security/biometric-card.html
[15] https://usa.visa.com/visa-everywhere/security/biometric-payment-card.html

developed by Fingerprint Cards[16] and Gemalto[17], to replace PIN/signature based user authentication and provide user convenience. The enrolled fingerprint template is stored on the card in a secure environment for additional security. See Figure 1.8.

1.3 Design of Automated Fingerprint Recognition Systems

In the early days of fingerprint use, primarily in law enforcement agencies, impressions were collected using off-line methods, *i.e.*, printer ink applied to subject's finger and then obtaining the impressions on ten-print cards (see Figure 1.10) which were then manually compared to a query fingerprint. These cards contain both *plain* and *rolled* impressions of all ten fingers[18]. While ten-print cards are still in use by some law enforcement agencies, most have moved to digital fingerprint acquisition via slap scanners.[19,20].

With the advancements in both fingerprint sensing technology and automated matching algorithms, ten-print fingerprint recognition has become extremely accurate and efficient. A typical recognition system contains the following two stages: *enrollment* and *recognition* (see Figure 1.9).

1. *Enrollment*: During this stage, an individual's fingerprint acquired using a fingerprint reader is processed to extract salient features and generate a *fingerprint template*. The template is then tagged with a unique user identifier for retrieval and is stored with associated metadata in a database, known as *reference*, *background*, *gallery*, or *enrollment* database.

2. *Recognition*: Depending on the application context, the recognition of an individual can be done to either validate the claimed identity (verification) or to establish the identity of an unknown individual (identification). In both cases, a fingerprint is acquired and processed to generate a template, known as *query* or *probe* template.

[16]https://www.fingerprints.com/solutions/payments/

[17]https://www.gemalto.com/financial/cards/emv-biometric-card

[18]A plain (or slap) fingerprint refers to an impression made by pressing a finger flat on a surface, and a rolled fingerprint is an impression made by rolling a finger from nail-to-nail in order to capture all of friction ridge details including sides.

[19]https://www.edo.cjis.gov/artifacts/standard-fingerprint-form-fd-258-1.pdf

[20]https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics/recording-legible-fingerprints

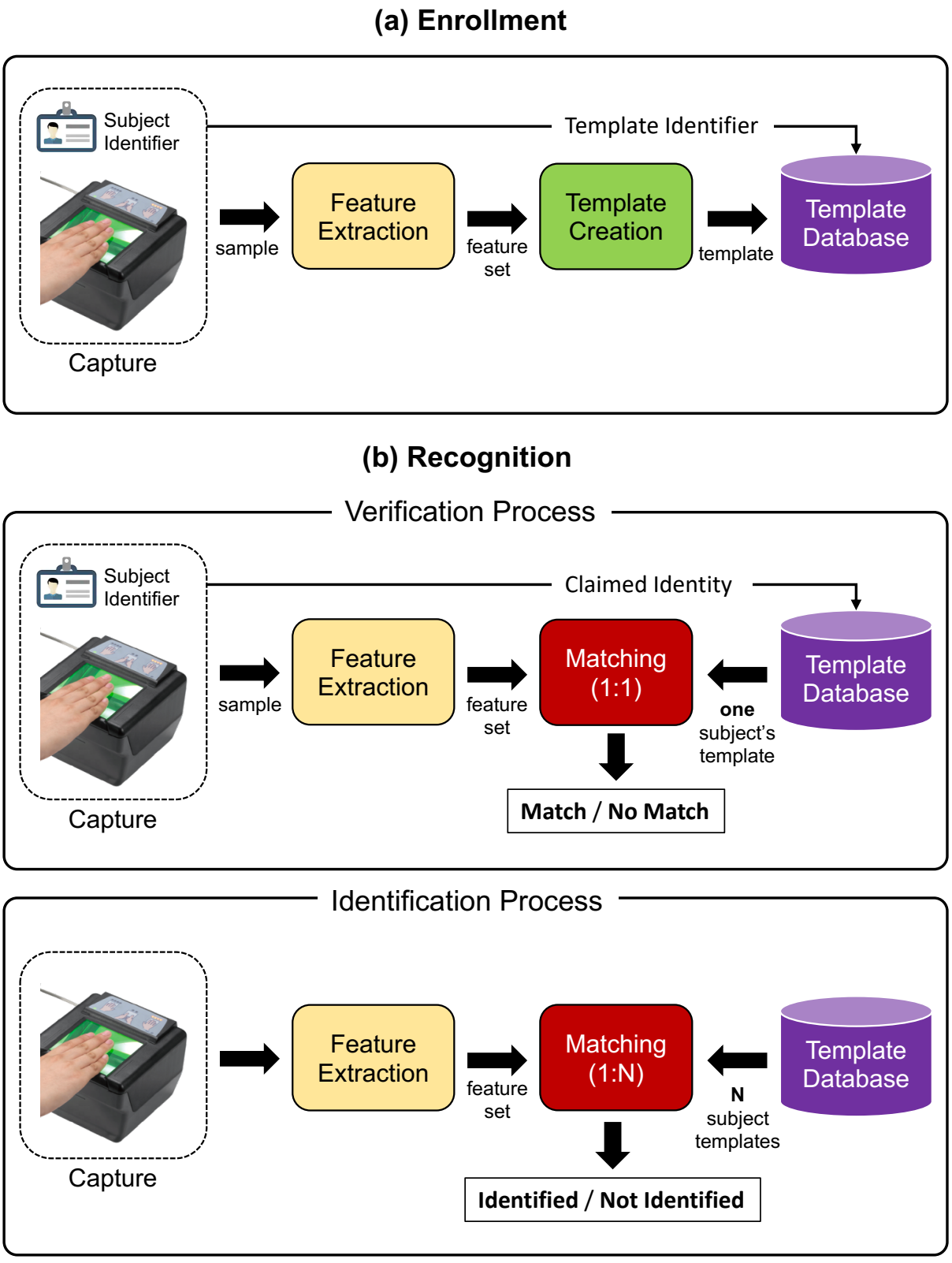


Figure 1.9 The two major stages of a fingerprint recognition system (a) enrollment and (b) recognition (verification or identification) are presented. These stages use the following modules: capture, feature extraction, template creation, matching, and template database. Image adapted from [104].

- *Verification*: In the verification scenario, the query template is accompanied by a user identifier (claim of identity) which is used to retrieve the enrolled template from the reference database. The system either accepts or rejects the submitted claim of identity by performing a one-to-one comparison between the query template and the retrieved reference template. Popular examples of this scenario include fingerprint-based access control and large-scale civil ID system (e.g. Aadhaar), where the user provides a unique ID (e.g. employee RFID card or Aadhaar 12-digit unique ID) and a fingerprint impression for authentication.
- *Identification*: In the identification scenario, no claim of identity is made. The goal of the system is to establish an identity of a subject by searching the entire reference database for a match. Therefore, a biometric system operating in the identification mode performs one-to-many comparisons to establish if the user is already enrolled in the database, and if so, returns the user identifier that matched. The system may also determine that the subject is not enrolled in the reference database. A common use-case of this scenario is a criminal investigation, where a fingerprint left at the crime scene is used to identify if the perpetrator is already enrolled in the database.

The enrollment, verification, and identification processes involved in fingerprint recognition make use of the following modules: (i) Fingerprint Acquisition, (ii) Feature Extraction, (iii) Template Database, and (iv) Matching.

1.3.1 Fingerprint Acquisition

The process of capturing the friction ridge details as a fingerprint impression for enrollment or recognition is known as *fingerprint acquisition*. It can be carried out in either a *controlled* or an *uncontrolled* manner. There are two controlled acquisition methods: (i) *off-line* methods such as applying ink on the fingertip and creating an inked impression by pressing (*i.e.*, plain/slap fingerprints) or rolling the fingertip (*i.e.*, rolled fingerprints) on paper, and (ii) *live-scan* methods

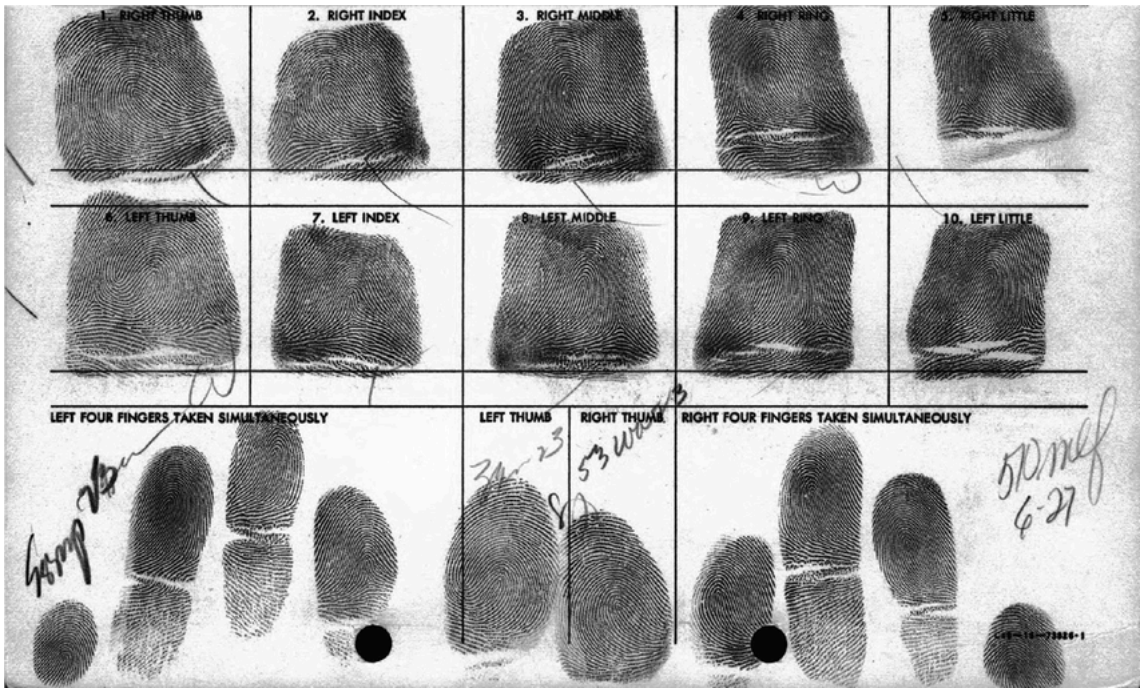


Figure 1.10 Tenprint card used by the FBI to collect fingerprint impressions of all ten fingers. The top two rows present the rolled impressions of all ten fingers, and the bottom row presents the plain/slap impressions in 4-4-2 pattern. Image reproduced from [83].

which utilize electronic fingerprint sensors[21] to acquire digital friction ridge impressions (see Figure 1.11). In both of these methods, the capture conditions are favorable with a cooperative subject, resulting in noise-free impressions on a clear background. Such impressions are known as *exemplar* fingerprints.

On the other hand, in the case of uncontrolled (or non-attended) fingerprint acquisition, there is no guarantee of the quality of acquired image. This is especially true for latent fingerprints at crime scenes which are routinely used by forensics agencies to find the culprit. Extremely important in forensic applications, latent fingerprints (also known as finger marks) are the friction ridge impressions unintentionally left on a surface touched by the fingertips. The oil secreted from the sebaceous glands in the skin gets deposited on a surface, such as glass, currency note, etc., touched by the finger. Depending on the characteristics of the surface, latents are enhanced and

[21] A fingerprint *reader* is a "black box" device, sold "as-is" by a commercial vendor, which typically contains an imaging *sensor* that acquires digital fingerprint images. However, in literature, the term *fingerprint sensor* is interchangeably used to imply a *fingerprint reader*.

(a) (b)

Figure 1.11 Two types of cooperative fingerprint acquisition methods: (i) off-line method using ink-on-paper technique, and (ii) live-scan method using an electronic fingerprint reader to capture a digital friction ridge impression.

"lifted" (acquired) using physical (*e.g.* dust with powder), chemical (*e.g.* ninhydrin treatment), and/or photographical (*e.g.* ultraviolet imaging) methods. Figure 1.12 presents the different types of fingerprint impressions, namely plain, slap, rolled, and latent fingerprints.

The most widely used form of fingerprint acquisition is using live-scan devices to acquire a digital fingerprint. The main parameters characterizing a digital fingerprint image are: resolution, area, number of pixels, geometric accuracy, contrast, and geometric distortion [104]. To ensure good quality of the acquired fingerprint impression and interoperability between various AFIS, the US Criminal Justice Information Services (CJIS) released a set of specifications[22] that regulate the quality and the format of both fingerprint images and FBI-compliant off-line/live-scan scanners, called *Appendix F*. Another less-stringent standard designed to support one-to-one fingerprint verification for single-finger capture devices in civilian applications, specifically for the Personal Identity Verification program, is *PIV-071006*.

1.3.1.1 Sensing Technologies

The ubiquitous use of fingerprint recognition in many consumer and government applications has led to the development of compact, high-resolution, and low-cost fingerprint sensing technologies. There are a number of live-scan sensing mechanisms (e.g., optical, solid-state, ultrasound, opti-

[22] https://www.fbibiospecs.cjis.gov/Certifications/FAQ

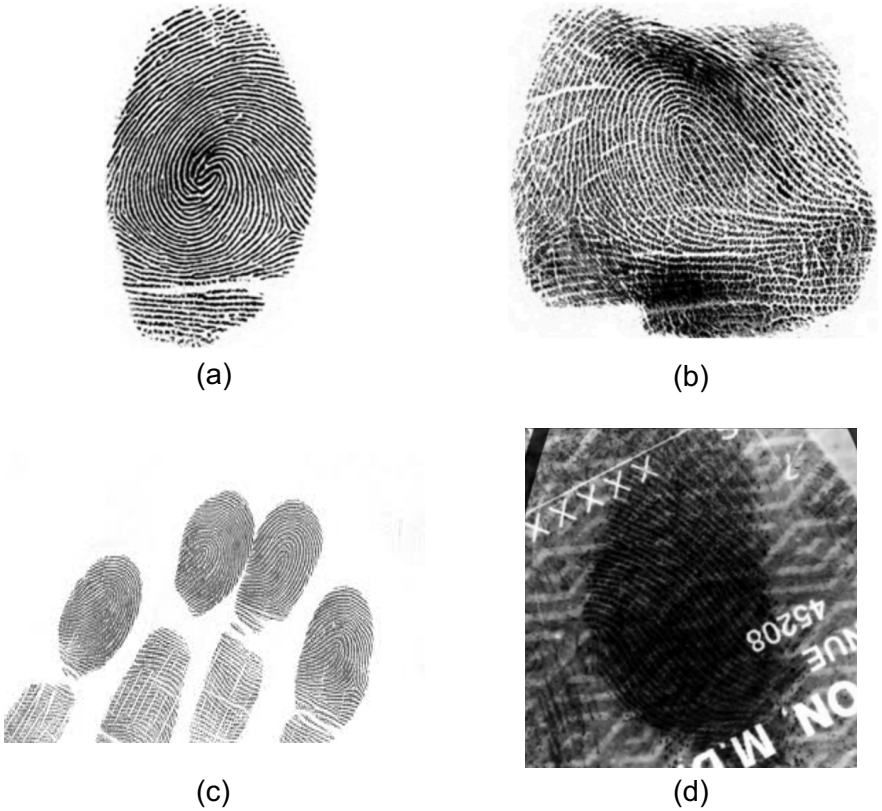
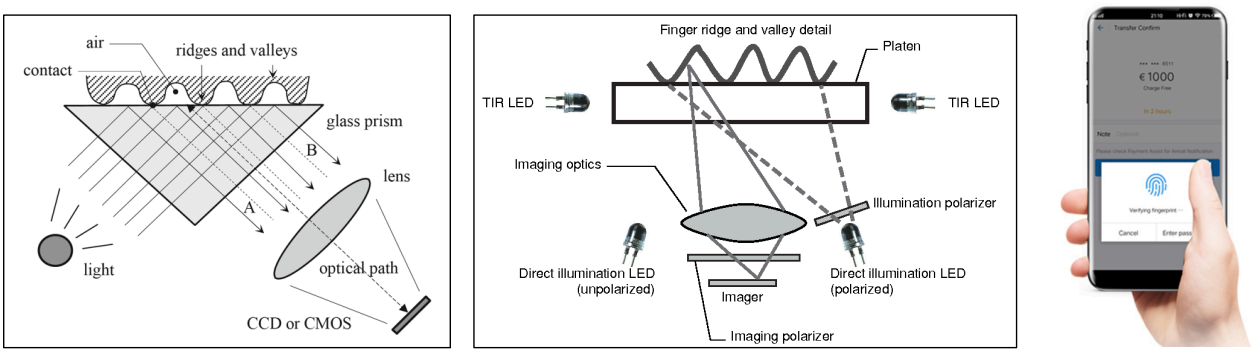


Figure 1.12 Different types of fingerprint impressions: (a) Plain/Flat, (b) Rolled, (c) Slap, and (d) Latent fingerprint.

cal coherence tomography, etc.) that can be used to detect the ridges and valleys present on the fingertip:

Optical: Fingerprint readers utilizing optical imaging are one of the most widely used readers in the commercial sector. Most optical readers operate on either the principle of Frustrated Total Internal Reflection (FTIR) or in a direct-view setup, where the camera/sensor directly captures the image of the finger. In the case of FTIR, the reader is typically an assembly of a glass prism, visible or infrared spectrum LEDs, and a CMOS or CCD sensor. The acquisition of a fingerprint involves the following steps: (i) the finger is placed on a glass prism, (ii) the finger surface is illuminated with LEDs, (iii) the incident light on the ridges is absorbed and that on the valleys undergo frustrated total internal reflection between the faces of glass prism to reach the sensor where the fingerprint is imaged [104]. In the case of direct-view imaging, the finger is placed on a glass platen, illuminated with LEDs, and the image is captured using a sensor placed below the platen.



(a) (b) (c)

Figure 1.13 Setup of optical fingerprint readers utilizing (a) a glass prism for Frustrated Total Internal Reflection (FTIR) of the incident light imaged using CCD or CMOS sensor, (b) direct-view multi-spectral setup employing polarized illumination of different wavelengths, and (c) an in-display optical sensing system for smartphones [65, 104, 138].

The image is processed to enhance the ridge-valley contrast. Some optical readers capture multiple images of the same finger, using different wavelengths (visible and near infrared) and different polarized conditions, which are fused together to produce a multi-spectral composite image. These images are robust to sub-optimal skin and ambient conditions [138]. However, one of the major limitations of optical readers is their bigger form factor, unlike solid-state capacitive readers, which has inhibited their use in small electronic devices such as smartphones. However, recent advancements have led to development of an *in-display* optical reader that is placed under the smartphone touchscreen [65] (see Figure 1.13). Figure 1.14 presents the different optical fingerprints sensors utilized in this thesis.

Solid-state: Solid-state sensing technology utilizes an array of mini-sensors to measure one of the following properties: (i) capacitance difference between ridges and valleys, (ii) pressure variations as finger interacts with sensor, or (iii) current generated on a pyro-electric sensor bed because of temperature differentials. Solid-state readers, because of their low cost and small size, are easily embeddable in hand-held devices such as laptops, tablets, and smartphones [104].

Ultrasound: The ultrasound sensing technology is based on sending acoustic signals towards the fingertip and sensing the echo response. The sensed echo response is processed to generate a depth profile of the fingertip, thereby providing the friction ridge structure. Ultrasound technology

CrossMatch Guardian 200
Slap Sensor
Sensing: FTIR
Resolution: 500 DPI

SilkID SLK20R
Single Finger Sensor
Sensing: FTIR
Resolution: 1,000 DPI

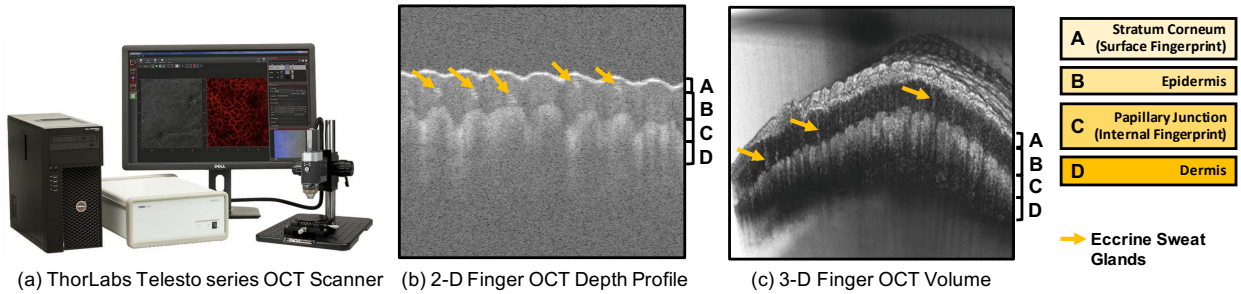
Lumidigm V302
Single Finger Sensor
Sensing: Direct View
Resolution: 500 DPI

Figure 1.14 Optical fingerprint sensors utilized in our experiments, namely CrossMatch Guardian 200, SilkID SLK20R, and Lumidigm V302.

is robust to oil, dirt, moisture, and other factors which may degrade the fingerprint image quality. Until recently, fingerprint readers utilizing ultrasound were expensive and large which inhibited their use in commercial applications. However, Qualcomm Inc. introduced an in-display ultrasound fingerprint sensor [37] which is now widely deployed in the Samsung smartphone series (Galaxy S10 onwards).

Optical coherence tomography (OCT): OCT [72] technology allows non-invasive, high-resolution, cross-sectional imaging of internal tissue microstructures by measuring their optical reflections. An optical analogue to Ultrasound [164], it utilizes low-coherence interferometry of near-infrared light ($900nm - 1325nm$). In an OCT scanner, a beam of light is split into a *sample arm*, *i.e.*, a unit containing the object of interest, and a *reference arm*, *i.e.*, a unit containing a mirror to reflect back light without any alteration. If the reflected light from the two arms are within coherence distance, it gives rise to an interference pattern representing the depth profile at a single point, also known as *A-scan*. Laterally combining a series of A-scans along a line can provide a cross-sectional scan, also known as *B-scan*. Stacking multiple B-scans together can provide a 3D

volumetric representation of the scanned object, or the object of our interest, *i.e.*, internal structure of a finger (see Figure 1.15).



(a) ThorLabs Telesto series OCT Scanner (b) 2-D Finger OCT Depth Profile (c) 3-D Finger OCT Volume

Figure 1.15 (a) Optical coherence tomography (OCT) scanner can be used to image the internal finger structure as (b) 2D and (c) 3D depth profile. Images reproduced from (a) [154], (b) IARPA ODIN Program (GCT-II) [123] and (c) [33].

1.3.2 Feature Extraction

The most evident characteristic of a fingerprint is its assemblage of interleaved ridges and valleys, where, typically, ridges are dark and valleys are bright. The fingerprint features (see Figure 1.16) are usually classified in a hierarchical order:

- *Level-1*: These global features, include fingerprint pattern type (arch, loop, whorl), singular points (cores, deltas), ridge orientation, and ridge spacing. These features are commonly used for indexing and fingerprint alignment, however, they cannot identify a fingerprint uniquely [104]. These features can be extracted by employing image processing techniques, detection of ridges with maximum curvature, or deep learning approaches [118, 134].

- *Level-2*: These local features refer to the salient points where ridges exhibit some discontinuity such as ridge endings and bifurcations, also known as *minutiae* points. In a rolled fingerprint, there can be over 100 minutiae, however, the spatial and angular coincidence of a small number of minutiae $(12-15)$ can be used to successfully match two fingerprints with high confidence [85]. The minimum recommended fingerprint image resolution to successfully extract minutiae points is 500 ppi. These features can be extracted using Gabor filters, dictionary-based methods, or CNN-based approaches [15, 22, 118].

- *Level-3*: These features include fingerprint characteristics at a very fine level of granularity such as sweat pores, incipient ridges, scars, creases, dots between the ridges, etc. These features provide additional uniqueness to a fingerprint, but require a minimum scanning resolution of 1000 ppi for successful extraction [79]. Primarily used by latent fingerprint examiners for manual comparison, these features are not commonly used in AFIS due to lack of robustness and high time requirements. However, recent developments in low-cost high resolution readers have led to the development of algorithms that utilize level-3 features for matching [17].

Prior to any fingerprint feature extraction, all fingerprint images typically undergo a preprocessing step (foreground extraction, enhancement, and/or alignment). In the case of latent fingerprints, where image quality is poor, preprocessing is a crucial step. State-of-the-art fingerprint commercial-off-the-shelf matchers (COTS) may utilize CNN-based methods for feature extraction similar to DeepPrint [44], and additional textural features at different scales [15].

1.3.3 Template Database

A fingerprint template is a set of features extracted from the fingerprint image of a user [96], such as variable-length minutiae-based features [] and fixed-length representation [44], . It is typically much smaller in size compared to the actual fingerprint image, providing faster processing time. International Standards Organization (ISO) defines standard template formats such as minutiae-based template standards ISO/IEC 19794-2 (2005) [22] for high interoperability, however, some commercial vendors may utilize a proprietary template format for high performance. The templates are associated with a unique user ID for retrieval and are stored in a database, referred to as *template database*.

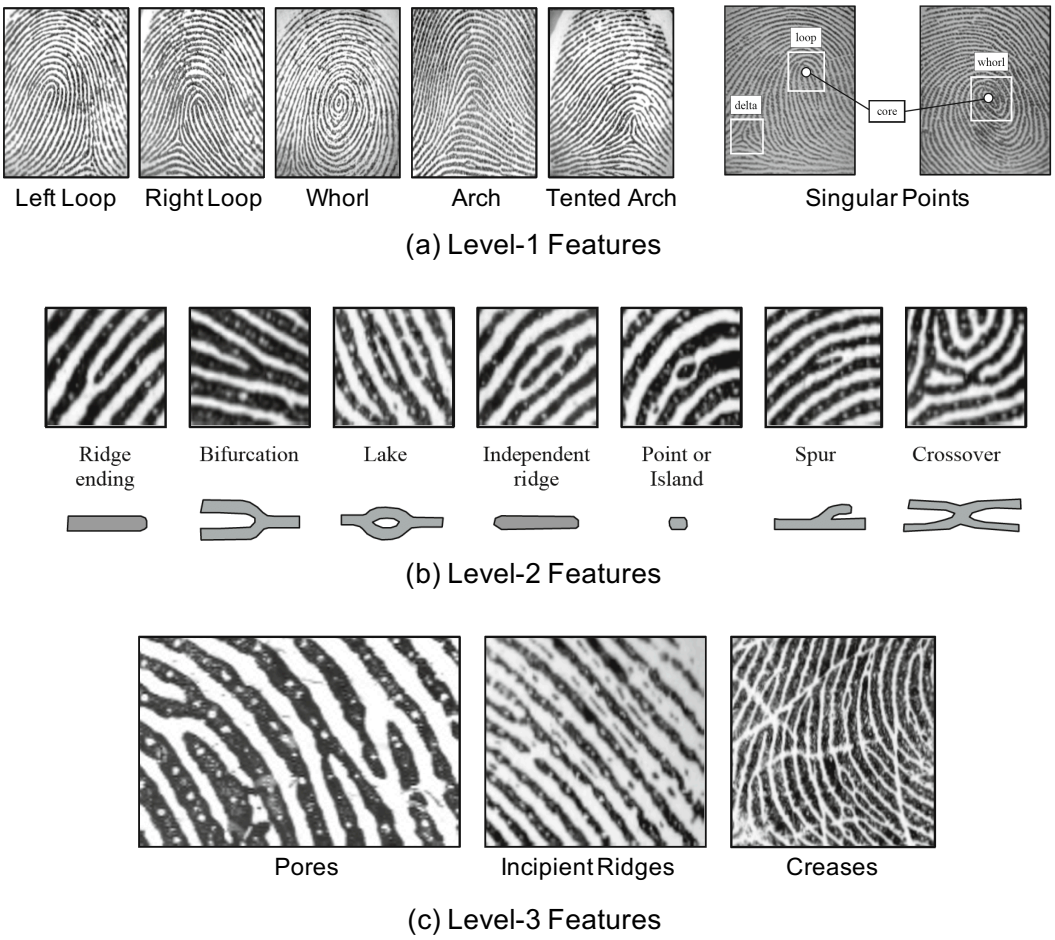


Figure 1.16 Fingerprint features are classified into three levels: (i) Level-1 features based on global fingerprint ridge pattern, (ii) Level-2 features based on local ridge characteristics, such as ridge endings, bifurcations, etc, and (iii) Level-3 features including finer details like sweat pores, incipient ridges and creases. Images reproduced from [104]

.

1.3.4 Fingerprint Matching

A fingerprint matching algorithm compares two given fingerprint templates and, typically, returns either a *similarity score*, say a value between 0 and 1 where a value close to 0 implies no similarity and close to 1 means very high similarity. Any match score above a specified threshold (t) is deemed as a successful match. A strict threshold (close to 1) provides high security (low false accepts) but results in poor user experience (due to high false rejects). Due to the large variability between different impressions of the same finger (intra-class variability), fingerprint matching is a difficult problem. Some of the main factors resulting in intra-class variations between fingerprints

include rotation, non-linear distortion, noise, displacement, partial overlap, pressure and skin conditions [104]. There are essentially three broad categories of fingerprint matching approaches:

- *Correlation-based matching*: This technique involves superimposing two fingerprint images and computing the correlation between the corresponding pixels for different alignments, rotations, and displacements. Due to the resource-intensive matching process, these techniques are not widely used.

- *Minutiae-based matching*: It is the most popular and widely deployed technique for fingerprint matching by both automated algorithms as well as fingerprint examiners. It involves finding the alignment between the reference minutiae set and the input query minutiae set that result in the maximum number of paired minutiae.

- *Non-minutiae feature based matching*: In the case of low-quality images, such as latent fingerprints, minutiae extraction is extremely difficult. This family of matching approaches may utilize either ridge pattern characteristics (e.g. local ridge frequency and orientation) or texture information using hand-crafted or deep learning methods [15, 17]. A fusion of minutiae-based and texture-based features can successfully improve the matching performance of latent fingerprints, including state-of-the-art deep-learning based methods with fixed length representation [44].

1.4 Challenges in Fingerprint Recognition

Fingerprint recognition is one of the most widely used methods for person recognition achieving a high level of matching accuracy and throughput in large-scale operational applications [161]. Despite tremendous improvements in the state-of-the-art [83, 104], fingerprint recognition encounters many remaining challenges and vulnerabilities.

1.4.1 Automatic Latent Fingerprint Recognition

Inadvertently left at crime scenes, latent fingerprints are one of the most crucial forms of evidence to identify or exclude a suspect in criminal investigations. In the current practice of matching latents, fingerprint examiners are expected to follow the Analysis, Comparison, Evaluation, and Verification (ACE-V) methodology [6]. In the "analysis" phase, latent prints are manually examined to perform a triage by assigning one of the following three values to a query latent: Value for Individualization (VID), Value for Exclusion Only (VEO) or No Value (NV). In the case of latents deemed to be "of value" (VID and VEO), the features in the latent are marked to search for their mates using an AFIS. In the "comparison" phase, the latent is manually compared side-by-side with the candidate mates retrieved from the exemplar database. In the "evaluation" phase, one of the following decisions is made about the latent in question: individualization, exclusion, or inconclusive[23]. Finally, in the "verification" phase, the decision made by the first examiner is confirmed by having a second examiner analyze the results independently.

Although the ACE-V methodology is widely accepted in the forensic community, the human subjectivity in the ACE-V process has raised concerns about its reliability and reproducibility. A notable case is the false accusation of Brandon Mayfield in the 2004 Madrid train bombing incident based on the incorrect match between Mayfield's exemplar fingerprint and the latent fingerprint lifted from the bomb site [124]. Along with the efforts to understand the human factors in latent fingerprint examination [25], standards and guidelines for latent examiners' practices have also been set up. As an example, Science Working Group on Friction Ridge Analysis, Study and Technology (SWGFAST) published standards to alleviate subjectivity involved in feature markups and decision makings among examiners [143]. Furthermore, with the growing caseload faced by

[23]Individualization refers to the decision on a pair consisting of a latent and an exemplar print indicating that the pair originates from the same finger based on a sufficient agreement between the two ridge patterns. Exclusion, on the other hand, is based on a sufficient disagreement between the two ridge patterns concluding that the pair did not originate from the same finger. An inconclusive decision is made when an examiner cannot make a decision of either individualization or exclusion due to insufficient ridge details or small corresponding area between latent and exemplar print [143].

forensic agencies, there is a need to develop methods for automatic and objective value assignment and matching for latents [17, 25].

1.4.2 Interoperability of Fingerprint Readers

Consider a fingerprint matching system that acquires fingerprint images using an optical reader during enrollment and a solid-state capacitive sensor during verification. Due to the variations in imaging technology, image resolution, sensor area, position of the sensor with respect to the user, and so on, the raw fingerprint images obtained from the two sensors will be different. This directly impacts the feature set extracted from the acquired images, and consequently, the match scores generated by the system.

Especially in the deployment of large-scale biometric projects, such as Aadhaar, one can not operate under the assumption that the fingerprint images to be compared will be obtained using the same sensor as it will restrict our ability to match fingerprint images originating from different sensors. Although progress has been made in the development of common data exchange formats and image quality standards[24] to facilitate the exchange of feature sets between vendors, very little effort has been invested in the actual development of algorithms and techniques to match these feature sets [137].

1.4.3 Vulnerabilities of an AFIS

While fingerprint recognition systems are deployed to protect an application from unauthorized access, the security of the system itself can be jeopardized implying no guarantee that the system will be completely secure. The fingerprint recognition system, like any other security system, is susceptible to a number of security threats as shown in Fig. 1.17. These system vulnerabilities may have adverse consequences such as intrusion by unauthorized users, denial-of-service to legitimate users, erosion of user privacy, or even identity theft. It must be emphasized that biometric system

[24]The ISO/IEC 19794-4 (2005) describes the manner in which a fingerprint image must be acquired and stored to maximize interoperability.

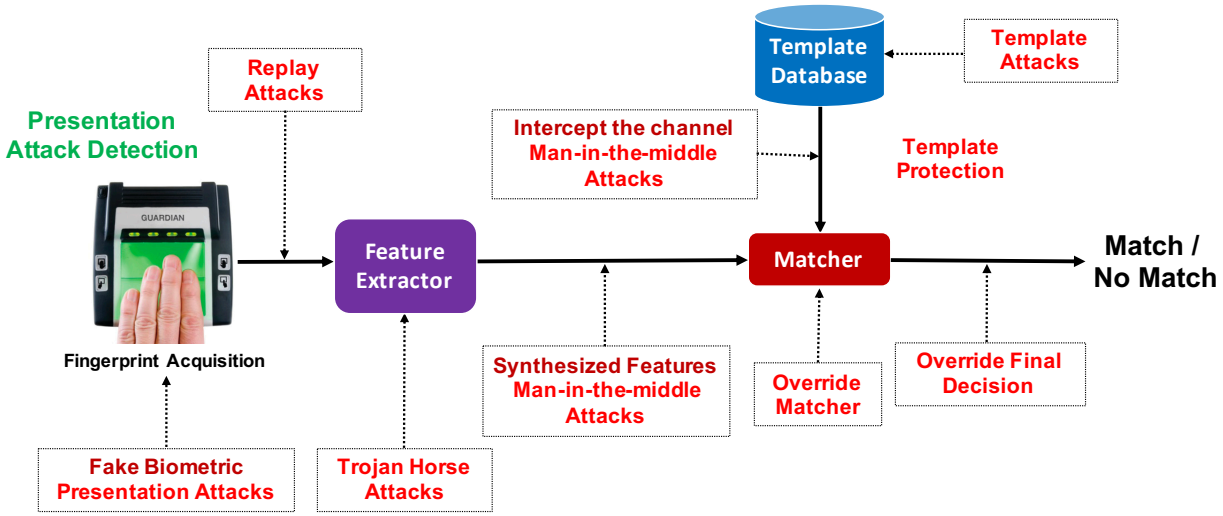


Figure 1.17 Different components in a fingerprint recognition system are vulnerable to various types of attacks shown in red. This thesis contributes towards addressing some of the challenges pertaining to presentation attack detection.

security and user privacy concerns are important public perception issues, which can potentially derail the success of a biometric system deployment unless they are addressed comprehensively.

While some of the typical security threats, such as replay and man-in-the-middle attacks, can be addressed by employing counter-measures taken from secure password-based authentication paradigms, the two main challenges specific to the domain of fingerprint recognition systems are (i) presentation attack detection (or liveness detection), and (ii) template protection.

1.4.3.1 Presentation Attack Detection

The ISO standard *IEC 30107-1:2016(E)* [74] defines presentation attacks as the *"presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system"*. These attacks can be realized through a number of methods including, but not limited to, use of (i) *gummy fingers* [108], *i.e.*, fabricated finger-like objects with accurate imitation of another individual's fingerprint ridge-valley structures, (ii) *2D or 3D printed fingerprint targets* [4, 5, 14], (iii) *altered fingerprints* [170], *i.e.*, intentionally tampered or damaged real fingerprint patterns to avoid identification, and (iv) *cadaver fingers* [105] (see Figure 1.18). Among these, fingerprint spoof attacks (*i.e.*, gummy fingers and printed targets) are the most common form

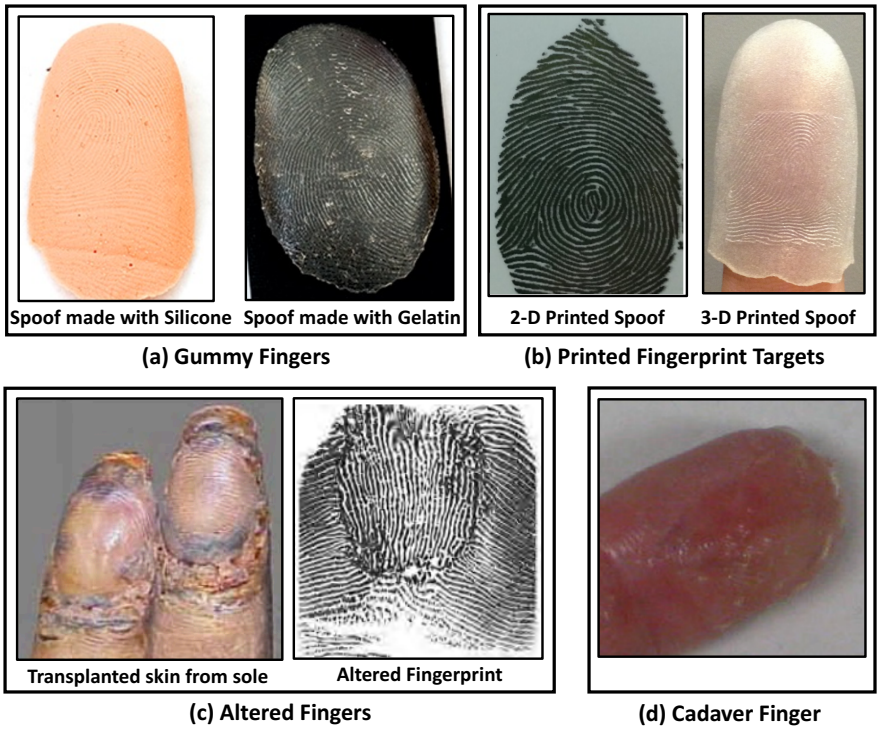


Figure 1.18 Fingerprint presentation attacks can be realized using (a) gummy fingers [57, 108], (b) 2D or 3D printed fingerprint targets [4,5,14], (c) altered fingers [170], or (d) cadaver fingers [105].

of presentation attacks, with a multitude of fabrication processes ranging from basic *molding and casting* to utilizing sophisticated 2D and 3D printing techniques [4, 5, 14, 42, 108]. Figure 1.19 illustrates a simple molding and casting procedure to create a presentation attack instrument using gelatin.

Unlike gummy fingers, altered or obfuscated fingerprints are real fingers whose ridge structure has been severely altered by abrading, burning, cutting, or performing surgery on fingertips (see Figure 1.20). The purpose of fingerprint obfuscation is to conceal one's identity in order to evade AFIS, especially for criminal re-identification and international border crossing [117, 170]. To be useful in practice, presentation attack detection schemes must recognize such attempts in real-time and with high accuracy without causing too much inconvenience to legitimate users.

Figure 1.19 Example procedure to create an artificial fingerprint directly from a live finger. Plastic is used to create the mold and gelatin is used as the casting material. Image reproduced from [105].

1.4.3.2 Template Protection

The other major challenge is the system security and user privacy issues arising from the leakage of fingerprint template information due to attacks on the template database. It has been shown that a fingerprint image can be reconstructed given the minutiae template [13]. Additionally, with the growing number of hacking attempts on large-scale central repositories containing biometric templates such as law enforcement and national ID databases[25], there is an urgent need to prevent leakage of personal user information. With more than 1.24 billion enrollments in India's national ID program, Aadhaar, the central repository houses more than 12.4 billion fingerprint templates [159]. Keeping the biometric templates in a centralized repository makes it prone to Distributed Denial-of-Service (DDOS) attacks affecting the availability during valid authentic attempts. In January 2018, it was reported that for Rs. 500 (under $\$10$) one can illegally obtain access to any person enrolled in the Aadhaar database within 10 minutes[26].

[25]https://www.thenewsminute.com/article/aadhaar-data-stolen-i-t-grids-proves-uidais-main-database-can-be-breached-experts-100215

[26]https://www.tribuneindia.com/news/nation/rs-500-10-minutes-and-you-have-access-to-billion-aadhaar-details/523361.html

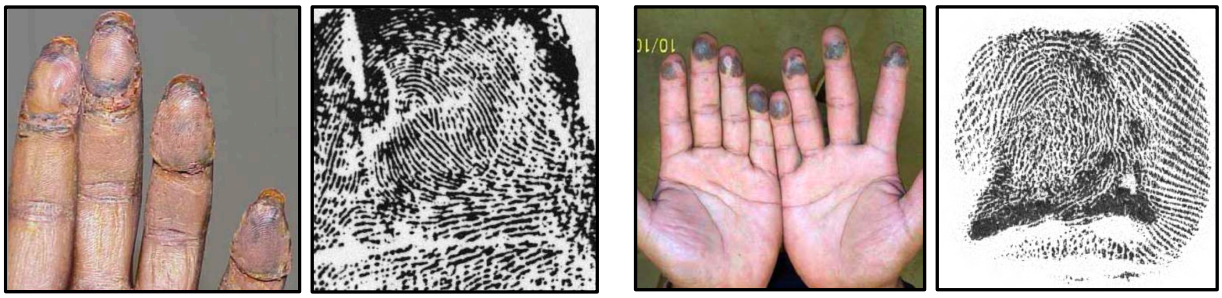


Figure 1.20 Example images of altered fingerprints. (a) Transplanted friction ridge skin from sole, and (b) fingers that have been bitten. Image source: [170]

In an operational scenario, typically, fingerprint templates are secured by using standard encryption techniques, e.g., AES, where the security of the template lies in knowledge of the decryption key. During authentication, templates are decrypted leaving them vulnerable to attacks. To overcome this, the templates are either stored and matched on device in a secure environment[27], or matched in the encrypted domain by employing homomorphic encryption [9, 44]. In literature, many template protection approaches have been proposed that aim to ensure non-invertibility, revocability, and non-linkability of templates while affording high recognition performance [82, 104]. However, there is still a need to bridge the gap between the theoretical proofs and the practical application of these approaches [44, 115].

1.5 Dissertation Contributions

The main contributions of this dissertation are as follows:

1. An accurate and robust deep learning-based fingerprint presentation attack detector (PAD), called *Fingerprint Spoof Buster*, utilizing local patches centered and aligned along fingerprint minutiae. Experimental results on publicly available datasets (LivDet 2011 - 2017), including intra-sensor, cross-material, cross-sensor, and cross-dataset scenarios, show that the proposed approach outperforms the state-of-the-art results published on these three datasets.

[27] https://support.apple.com/en-sg/HT204587

For example, in LivDet 2015 (2017), our algorithm achieves 99.03% (95.91%) average accuracy over all sensors compared to 95.51% (95.25%) achieved by the LivDet 2015 (2017) winner [113, 172].

2. A graphical user interface which highlights the local regions of the fingerprint image as bonafide (live) or PA (spoof) for visual inspection. This is more informative than a single spoof score output by the traditional approaches for the entire fingerprint image.

3. An algorithm for detection and localization of fingerprint alterations (fingerprint obfuscation). The proposed approach achieves a state-of-the-art True Detection Rate (TDR) of 99.24% @ False Detection Rate (FDR) of 2% on an operational altered fingerprint database from a law enforcement agency.

4. A light-weight version of the PAD, called *Fingerprint Spoof Buster Lite*, as an Android application that can run on a commodity smartphone (Samsung Galaxy S8) without a significant drop in performance (from TDR = 95.7% to 95.3% @ FDR = 0.2%) in under 100ms.

5. An interpretation of cross-material (generalization) performance of the proposed PAD by (i) evaluating Fingerprint Spoof Buster against unknown PAs by adopting a leave-one-out protocol; one material is left out from training set and is then utilized for testing, (ii) utilizing 3D t-SNE visualizations of the bonafide and PA samples in the deep feature space, (iii) investigating the PA material characteristics (two optical and two physical properties) and correlating them with their cross-material performances, to identify a representative set of PA materials that should be included during training to ensure a high generalization performance.

6. A style transfer-based wrapper, called *Universal Material Generator* (UMG), to improve the generalization performance of any PA detector against novel PA fabrication materials that are unknown to the system during training. The proposed wrapper is shown to improve the average generalization performance of Fingerprint Spoof Buster from TDR of 75.24% to

91.78% @ FDR = 0.2% when evaluated on a large-scale dataset of $5,743$ live and $4,912$ PA images fabricated using 12 materials. It is also shown to improve the average cross-sensor performance from 67.60% to 80.63% when tested on LivDet 2017 dataset, alleviating the time and resources required to generate large-scale PA datasets for new sensors.

7. A dynamic PAD solution utilizing a sequence of local patches centered at detected minutiae from ten color frames captured in quick succession (8 fps) as the finger is presented on the sensor. We posit that the dynamics involved in the presentation of a finger, such as skin blanching, distortion, and perspiration, provide discriminating cues to distinguish live from spoofs. The proposed approach improves the spoof detection performance from TDR of 99.11% to 99.25% @ FDR = 0.2% in known-material scenarios, and from TDR of 81.65% to 86.20% @ FDR = 0.2% in cross-material scenarios.

8. A PAD solution utilizing the ridge-valley depth-information of finger skin, including internal fingerprint (papillary junction) and sweat (eccrine) glands, sensed by the optical coherent tomography (OCT) fingerprint technology. Our proposed solution achieves a TDR of 99.73% @ FDR of 0.2% on a database of $3,413$ bonafide and 357 PA OCT scans captured using THORLabs Telesto series spectral-domain fingerprint reader. We also identify the regions in the OCT scan patches that are crucial for fingerprint PAD detection.

Chapter 2

Fingerprint Presentation Attack Detection

This chapter addresses the problem of developing an accurate, robust, and efficient solution for detecting fingerprint presentation attacks. Specifically, we propose a deep learning-based approach, called *Fingerprint Spoof Buster*, utilizing local patches centered and aligned using fingerprint minutiae to train deep convolutional neural networks (CNNs). Experimental results on publicly-available LivDet datasets, an operational altered fingerprint database, three large-scale government controlled evaluations as part of the IARPA ODIN project, and two in-house collected PA datasets containing more than $20,000$ images (12 PA materials) show that the proposed approach achieves state-of-the-art performance in fingerprint presentation attack detection for intra-sensor, cross-material, cross-sensor, and cross-dataset testing scenarios.

In order to understand the decision made by CNN, we have developed a graphical user interface that allows the operator to visually examine the local regions of the fingerprint image highlighted as bonafide (live) or PA (spoof/altered), instead of relying on a single spoof score as output by competing PAD approaches. We also present a light-weight version of the proposed PAD, called *Fingerprint Spoof Buster lite*, as an Android app that can run on a commodity smartphone (Samsung Galaxy S8) without a significant drop in PAD performance (from TDR = 95.7% to 95.3% @ FDR = 0.2%) in under 100ms.

2.1 Introduction

With the proliferation of automated fingerprint recognition systems in many applications, including mobile payments, international border security, and national ID, the vulnerability of the system security to *presentation attacks* is of growing concern [30, 107, 123]. These attacks can be realized through a number of methods including, but not limited to, (i) *gummy fingers* [108], *i.e.*, fabricated finger-like objects with an accurate imitation of one's fingerprint to steal their identity, (ii) *2D or 3D printed fingerprint targets* [5, 14, 42], (iii) *altered fingerprints* [152, 170], *i.e.*, intentionally tampered or damaged real fingerprint patterns to avoid identification, and (iv) *cadaver fingers* [105]. Among these, fingerprint spoof attacks (*i.e.*, gummy fingers and printed targets) are the most common and easiest to launch form of presentation attacks, with a multitude of fabrication processes ranging from basic *molding and casting* to utilizing sophisticated 2D and 3D printing techniques [4, 5, 14, 42, 108].

It has been reported that commonly available and inexpensive materials, such as gelatin, silicone, play-doh, etc., have been utilized to fabricate high fidelity fingerprint spoofs which are capable of bypassing a fingerprint recognition system. See Figs. 2.1 and 2.2. In March 2013, a Brazilian doctor was arrested for using spoof fingers made of silicone to fool the biometric attendance system at a hospital in Sao Paulo[1]. In another incident, in Sept. 2013, shortly after Apple released iPhone 5s with inbuilt TouchID fingerprint technology, Germany's Chaos Computer Club[2] hacked its capacitive sensor by utilizing a high resolution photograph of the enrolled user's fingerprint to fabricate a spoof fingerprint with wood glue. In July 2016, researchers at Michigan State University unlocked a fingerprint secure-smartphone using a 2D printed fingerprint spoof to help police with a homicide case[3] [14]. In March 2018, a gang in Rajasthan, India, was arrested for spoofing the biometric attendance system, using glue casted in wax molds, to provide proxies for

[1] http://www.bbc.com/news/world-latin-america-21756709

[2] http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid

[3] http://statenews.com/article/2016/08/how-msu-researchers-unlocked-a-fingerprint-secure-smartphone-to-help-police-with-homicide-case

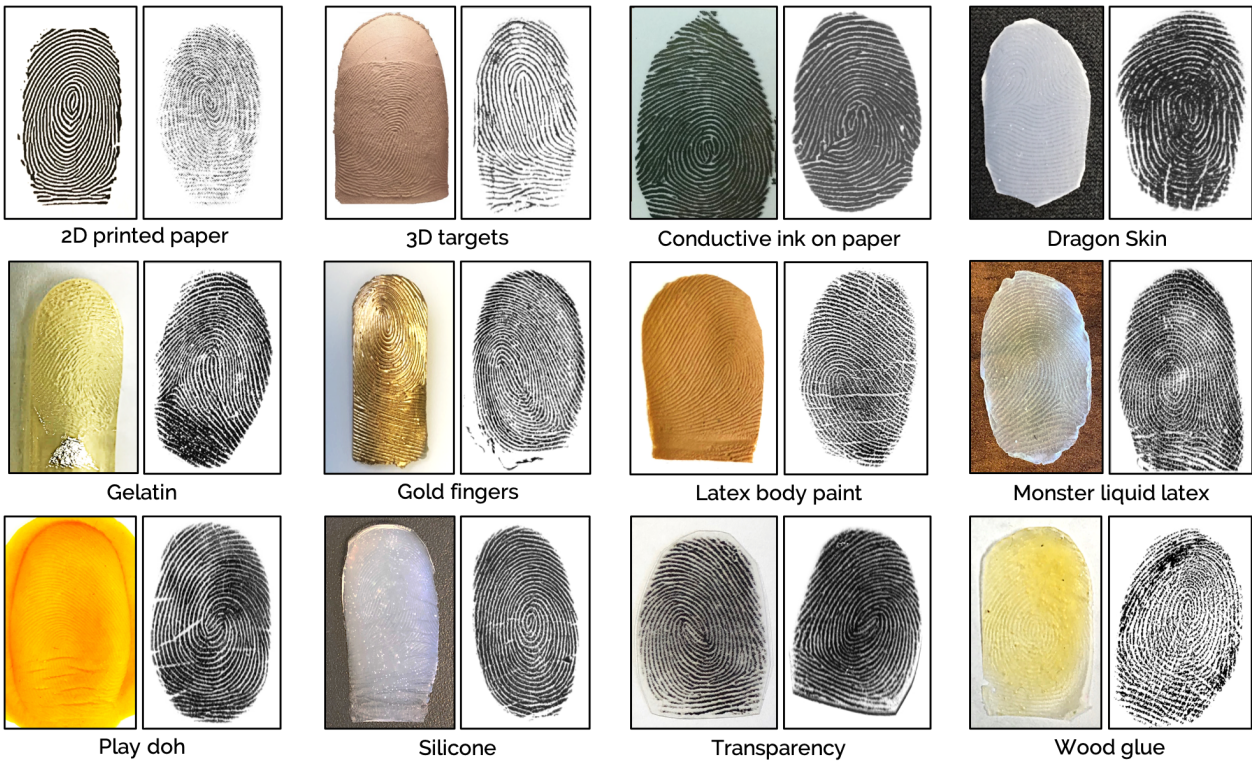


Figure 2.1 Fingerprint spoof attacks can be realized using various readily available fabrication materials, such as PlayDoh, WoodGlue, Gelatin, etc. For each of the image pairs, the left image presents the actual spoof specimen while the right image presents the grayscale fingerprint impression captured of that spoof on a CrossMatch Guardian 200 fingerprint reader.

a police entrance exam[4]. As recently as April 2019, a Galaxy S10 owner with the assistance of a 3D printer and a photo of his own fingerprint was able to spoof the ultrasonic in-display fingerprint sensor on his smartphone[5]. Other similar successful spoof attacks have been reported showing the vulnerabilities of fingerprint biometric systems deployed in various applications[6,7]. It is likely that a large number of these attacks are never detected and hence not reported.

Another form of presentation attacks include intentional fingerprint alteration, known as *altered fingerprints* (see Figs. 1.20 and 2.10), in an attempt to obfuscate the true identity to evade law enforcement AFIS [36]. Cases of tampering with fingerprints to evade detection in criminal cases were reported as early as 1935. Cummins [31] reported 3 cases of fingerprint alterations and

[4]https://www.medianama.com/2018/03/223-cloned-thumb-prints-used-to-spoof-biometrics-and-allow-proxies-to-answer-online-rajasthan-police-exam/

[5]https://www.reddit.com/r/galaxys10/comments/b97ur8/i_attempted_to_fool_the_new_samsung_galaxy_s10s/

[6]http://fortune.com/2016/04/07/guy-unlocked-iphone-play-doh/

[7]https://srlabs.de/bites/spoofing-fingerprints/

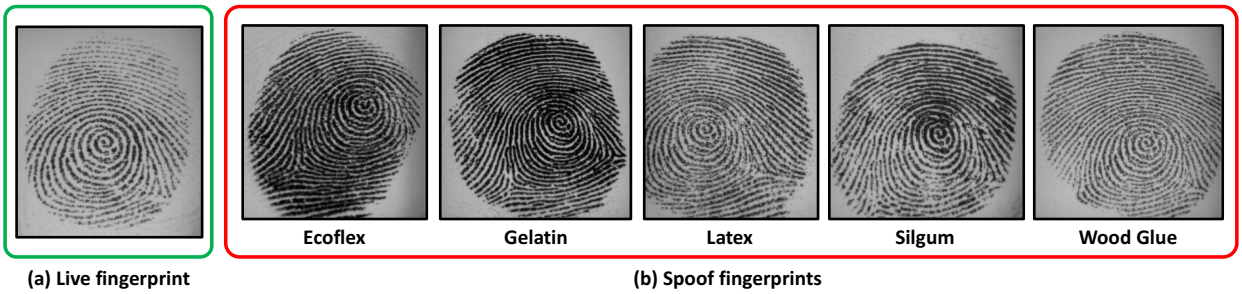


Figure 2.2 Visual comparison between (a) a live fingerprint, and (b) the corresponding spoofs (of the same finger) made with different materials. Images are taken from LivDet-2011 dataset (Biometrika sensor) [167]. Our method can successfully distinguish between live and spoof fingerprints. The spoofness score for live fingerprint is 0.00, and for spoof fingerprints the scores are 0.95, 0.97, 0.99, 0.99, and 0.95 for Ecoflex, Gelatin, Latex, Silgum, and Wood Glue, respectively.

presented images of before and after alterations. In recent years, border crossing applications have been targeted by altered fingerprint attacks. In 2009, ABC news reported that Japanese officials arrested a Chinese woman who took "a particularly extreme measure" to evade detection [117]. The Chinese woman had paid a plastic surgeon to swap fingerprints between her right and left hands. Patches of skin from her thumbs and index fingers were reportedly removed and then grafted onto the ends of fingers on the opposite hand. As a result, her identity was not detected when she re-entered Japan illegally. In 2014, the FBI identified 412 records in its IAFIS which indicated deliberate fingerprint alterations [121]. In 2018, Business Insider reported that Eduardo Ravelo, who was added to the FBI's 10 Most Wanted list in October 2009, was believed to have had plastic surgery to alter his fingerprints to evade authorities [122]. Therefore, presentation attack detection (PAD) is of utmost importance, especially in an unsupervised scenario (e.g., authentication on a smartphone, secure facility access, self check-in kiosks at airports) where the fingerprint presentation by a user is typically not monitored.

Table 2.1 Performance comparison (Average Classification Error [%]) of software-based spoof detection studies on LivDet 2011, 2013, 2015, and 2017 competition datasets. Since different competition databases utilize different fingerprint readers (optical / thermal / capacitive), spoof materials, and modes of data collection (cooperative/uncooperative), a direct performance comparison between different databases will not be a fair comparison.

Study	Approach	LivDet 2011	LivDet 2013*	LivDet 2015	LivDet 2017
Approaches utilizing hand-engineered features					
Ghiani et al., 2012 [56]	Local Phase Quantization (LPQ)	11.1	3.0	N/A	N/A
Gragniello et al., 2013 [60]	Weber Local Descriptor (WLD)	7.9	N/A	N/A	N/A
Ghiani et al., 2013 [55]	Binarized Statistical Image Features (BSIF)	7.2	2.1	N/A	N/A
Gragniello et al., 2015 [61]	Local Contrast-Phase Descriptor (LCPD)	5.7	1.3	N/A	N/A
Deep learning-based approaches					
Nogueira et al., 2016 [119]	Transfer Learning + CNN-VGG + Whole Image	4.5	1.1	4.5	N/A
Pala et al., 2017 [126]	Custom CNN with triplet loss + Randomly selected local patches	3.33	0.58	N/A	N/A
Zhang et al., 2019 [172]	CNN with residual blocks + Center of gravity-based local patches	N/A	1.74	3.18	4.75
Proposed Approach	CNN-MobileNet-v1 + Minutiae-based local patches	**1.67**	**0.25**	**0.97**	**4.56**

*LivDet 2013 includes results for Biometrika and Italdata sensors.

2.2 Related Work

2.2.1 Studies on Fingerprint Spoof Detection

The various spoof detection approaches proposed in the biometrics literature can be broadly classified into (i) hardware-based and (ii) software-based solutions [107]. In the case of hardware-based approaches, the fingerprint readers are augmented with sensor(s) which detect characteristics of vitality, such as blood flow, thermal output, heartbeat, odor, and skin distortion [2, 8, 94]. Additionally, special types of fingerprint sensing technologies have been developed for imaging the sub-dermal friction ridge surface based on multi-spectral [136, 138], short-wave infrared [156] and optical coherent tomography (OCT) [29, 111]. A low-cost "Build-It-Yourself" open-source fingerprint reader, called RaspiReader, uses two cameras to provide complementary streams (direct-view and FTIR) of images for spoof detection [43]. Ultrasound-based in-display fingerprint readers developed for smartphones by Qualcomm Inc. [1] utilize acoustic response characteristics for spoof detection.

Table 2.2 Related work on altered fingerprint detection. There is no public-domain altered fingerprint database available in the literature.

Source	Method	Dataset	Performance
Feng, Jain and Ross [48]	orientation field	1,976 simulated altered fingerprints	92% detection rate at false positive rate of 7%
Tiribuzi et al. [155]	minutiae density maps and orientation entropies	1000 genuine and synthetic altered fingerprints	90.4% classification accuracy
Yoon et al. [170]	orientation field and minutiae distribution	4,433 operational altered fingerprints from 270 subjects	70.2% detection rate at false positive rate of 2.1%
Ellingsgaard and Busch [40, 41]	orientation field and minutia orientation	116 altered and 180 unaltered from various sources	92.0% detection rate at false positive rate of 2.3%
Proposed Approach	input image and minutiae-based patches; CNN models	4,815 altered and 4,815 bonafide fingerprints from 270 subjects	99.24% detection rate at false positive rate of 2%

In contrast, software-based solutions extract salient features from the captured fingerprint image (or a sequence of frames) for separating live and spoof images. The software-based approaches in the literature are typically based on (i) anatomical features (e.g. pore locations and their distribution [142]), (ii) physiological features (e.g. perspiration [106]), and (iii) texture-based features (e.g. Weber Local Binary Descriptor (WLBD) [165], SIFT [59]. Most state-of-the-art approaches are learning-based, where the features are learned by training convolutional neural networks (CNN) [23, 24, 26, 87, 119, 126, 156, 172]. See Table 2.1.

2.2.2 Studies on Altered Fingerprint Detection

Detection of altered fingerprints is of high value to law enforcement and homeland security agencies to prevent known criminals (in the government watchlist) from evading the AFIS at border crossings and illegally entering the country. Existing approaches for detecting fingerprint alteration have primarily explored hand crafted features to distinguish between altered and bonafide fingerprints. Feng et al. [48] trained an SVM to detect irregularities in ridge orientation field and reported a 92% detection rate at a false positive rate of 7% on a database of $1,976$ simulated altered fingerprints. Tiribuzi et al. [155] combined the minutiae density maps and the orientation entropies of the ridge-flow to identify the altered fingerprints. They reported a 90.4% classification accuracy on a dataset of $1,000$ genuine and synthetic altered fingerprints. Yoon et al. [170] utilized the orientation field and minutiae distribution to detect altered fingerprints. Their method was

tested on a database of $4,433$ altered fingerprints from 270 subjects, providing for 70.2% correctly identified altered fingerprints at a false positive rate of 2.1%. Ellingsgaard and Busch in [40, 41] discuss methods for automatically detecting altered fingerprints based on analysis of two different local characteristics of a fingerprint image: identifying irregularities in the pixel-wise orientations, and examining minutia orientations in local patches. They further suggest that alteration detection should be included into standard quality measures of fingerprints. Beyond detection of altered fingerprints, Yoon et al. [170] investigated feasibility of an AFIS to link altered fingerprints to their pre-altered mates. Table 2.2 summarizes previous work in altered fingerprint detection. All the existing methods are based on examining irregularities in orientation flow or minutia maps based on *hand-engineered features*.

The proposed approach (Section 2.4) uses a deep learning technique to learn and evaluate salient features in the altered fingerprints, and classify input fingerprint images into two classes: bonafide or altered fingerprints. In the case of altered fingerprint, the proposed approach localizes the regions of a fingerprint that are altered. This can be utilized to assess the *fingerprintness* of an input image [170], such that bonafide fingerprints (or bonafide regions) produce a high score and altered fingerprints (or altered regions) produce a low score.

2.3 Fingerprint Spoof Buster

A series of fingerprint Liveness Detection (LivDet) competitions have been held since 2009 to advance state-of-the-art and benchmark the proposed anti-spoofing solutions [57]. The best performer in the LivDet 2015 [113], Nogueira et al. [119], utilized transfer learning, where deep CNNs originally designed for object recognition and pre-trained on ImageNet database [140], were fine-tuned on fingerprint images to differentiate between live and spoof fingerprints. In their approach, the networks were trained on whole fingerprint images resized to 227×227 pixels for VGG [147] and 224×224 pixels for AlexNet [93] as required by these networks. However, there are three disadvantages of using this approach: (i) fingerprint images from some of the sensors used in LivDet

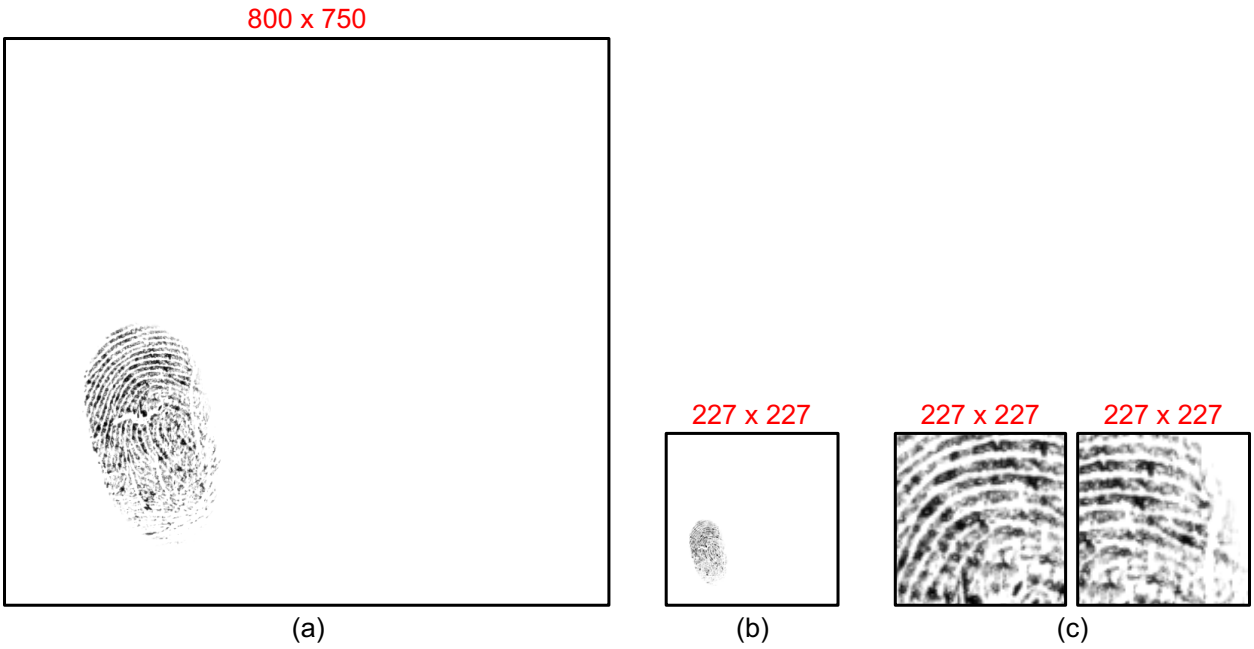


Figure 2.3 A live fingerprint image (from LivDet 2015 dataset) captured using CrossMatch L Scan Guardian in its (a) original dimensions (800×750), and (b) resized to 227×227. A direct downsizing of the fingerprint image may result in the friction ridge area occupying less than 10% of the original image size, leading to significant loss of discriminatory information. Instead, local patches (96×96 upscaled to 227×227), as shown in (c), provide salient cues to differentiate a spoof fingerprint from live fingerprint.

datasets, such as Crossmatch L Scan Guardian (800×750), have a large blank area ($\geq 50\%$) surrounding the friction ridge region. Directly resizing these images, from 800×750 to 227×227, eventually results in the friction ridge area occupying less than 10% of the original image size (see Figure 2.3); (ii) resizing a rectangular image of size, say $w \times h$, to a square image, say $p \times p$, leads to different amounts of information retained in the two spatial image dimensions; (iii) downsizing an image, in general, leads to a significant loss of discriminatory information.

It is important to consider various sources of noise involved in the spoof fabrication process itself that can introduce some artifacts, such as missing friction ridge regions, cracks, air bubbles, etc., in the spoofs. The primary consequence of such artifacts is the creation of spurious minutiae in the fingerprint images sensed from spoofs. The local regions around these spurious minutiae can, therefore, provide salient cues to differentiate a spoof fingerprint from live fingerprints (see Fig. 2.4). We utilize this observation to train a two-class CNN using local patches around the ex-

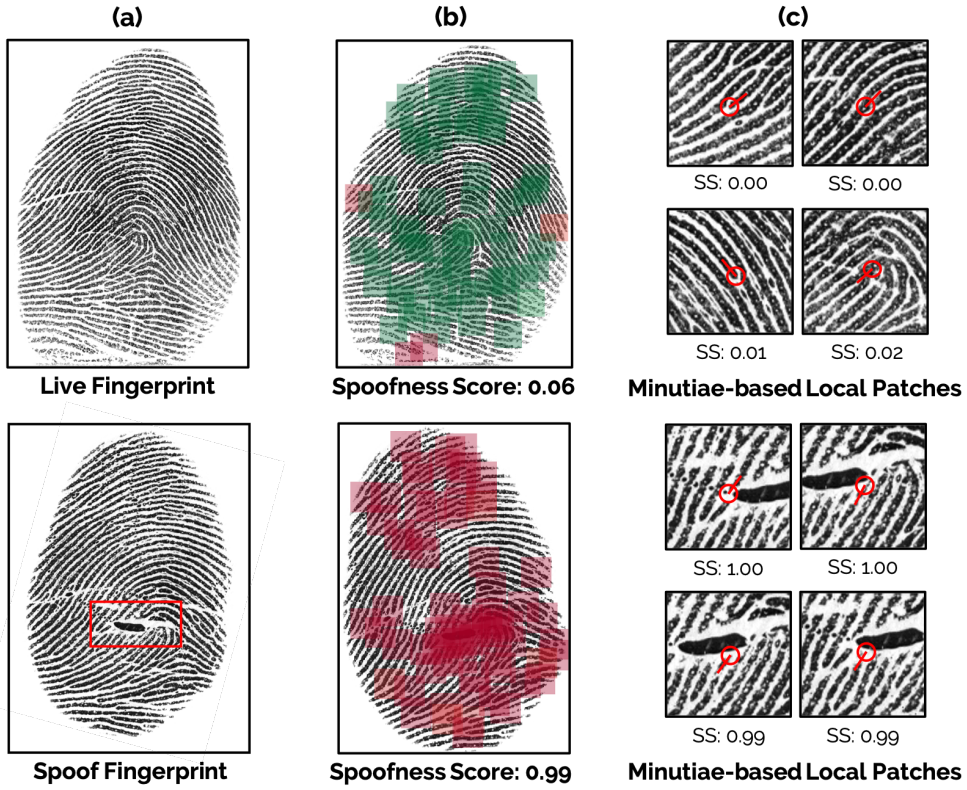


Figure 2.4 (a) Example of a live fingerprint and the corresponding spoof fingerprint with the artifacts introduced in the spoofs highlighted in red. (b) Local regions highlighted as green (live) and red (spoof) by evaluating all minutiae-centered local patches (96×96). (c) A subset of minutiae-based local patches along with their individual spoofness scores. The images are taken from MSU Fingerprint Presentation Attack Dataset (MSU-FPAD) - CrossMatch Sensor and the spoof material used is Silicone (Ecoflex). The spoofness scores output by the proposed approach for the live and spoof fingerprints are 0.06 and 0.99, respectively. (Best viewed in color)

tracted minutiae, as opposed to the whole fingerprint images or randomly selected local patches, to design a fingerprint spoof detector. In this section, we will show that the proposed approach, called *Fingerprint Spoof Buster*, is more robust to novel fabrication materials than earlier approaches that utilize the whole image [119] or randomly selected local patches [126].

The proposed approach for spoof detection, utilizing local patches of size $p \times p$, ($p = 96$), centered at minutiae, (i) circumvents the previously mentioned drawbacks of downsizing whole fingerprint images to train the CNNs, (ii) provides large amount of data (an average of 48 patches/fingerprint image) to train the deep CNN architectures without overfitting, (iii) learns salient textural information from local regions, robust to differentiate between spoof and live fin-

gerprints, and (iv) provides a fine-grained analysis of the fingerprint images by localizing and highlighting spoof regions. The output of the CNN is a confidence score in the range $[0, 1]$, defined as *Spoofness Score*; the higher the spoofness score, the more likely the image patch is extracted from a spoof fingerprint. For a given image, the spoofness scores corresponding to the minutiae-based local patches are averaged to generate the global spoofness score for the input image. A fusion of CNN models trained on multi-scale patches (ranging in size from from 64×64 to 128×128), centered and aligned using minutiae, is shown to further boost the spoof detection performance.

We also optimize Fingerprint Spoof Buster to reduce memory and computation requirements by (i) K-means clustering of minutiae points followed by weighted fusion to reduce the required number of local patches to be evaluated, and (ii) modifying the MobileNet-v1 network architecture and quantization of model weights to reduce the required computations and perform byte computations instead floating point arithmetic. Consequently, a light-weight version of the PAD (3.2 MB), called *Fingerprint Spoof Buster Lite*, is developed as an Android application that can run on a commodity smartphone without a significant drop in PAD performance in under 100ms. The main contributions of this chapter are enumerated below:

- Utilized fingerprint domain-knowledge to design a robust fingerprint spoof detector, called Fingerprint Spoof Buster, where local patches centered and aligned using fingerprint minutiae are utilized for training a CNN model. This differs from other existing approaches which have generally used the whole input fingerprint image for spoof detection.

- Experimental results on publicly available datasets (LivDet 2011, 2013, 2015, and 2017), including intra-sensor, cross-material, cross-sensor, and cross-dataset scenarios, show that the proposed approach outperforms the state-of-the-art results published on these three datasets. For example, for LivDet 2015 (2017) dataset, our algorithm achieves 99.03% (95.91%) average accuracy over all fingerprint readers compared to 95.51% (95.25%) achieved by the LivDet 2015 (2017) winner [113, 172].

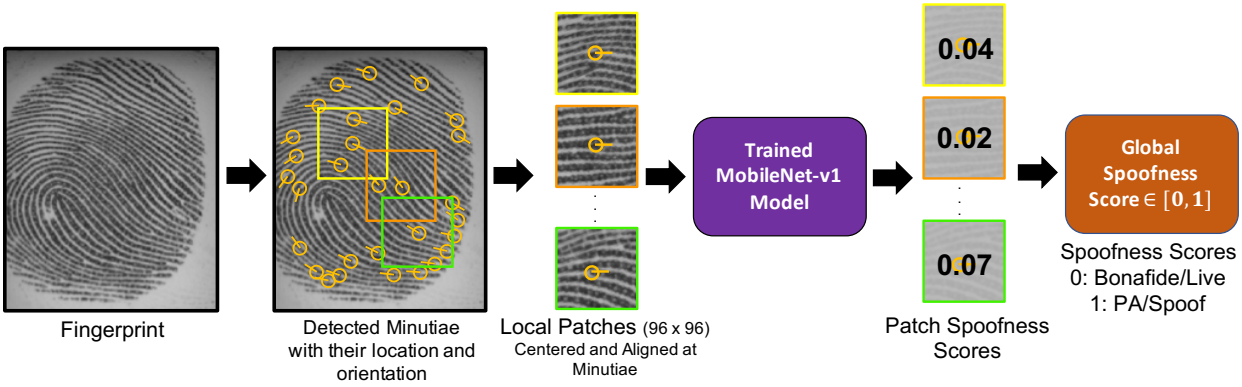


Figure 2.5 An overview of the proposed Fingerprint Spoof Buster [24], a state-of-the-art fingerprint PAD, utilizing CNNs trained on local patches centered and aligned using minutiae location and orientation, respectively. A total number of 30 minutiae are detected in the input fingerprint image.

- Collected two new fingerprint presentation attack datasets containing more than $20,000$ fingerprint (live and spoof) images, using two different fingerprint readers and over 12 different spoof fabrication materials. Experimental results on these two new datasets and three large-scale government test datasets as part of IARPA ODIN project are also presented. IARPA consider these results to be state-of-the-art[8].

- Developed a graphical user interface (GUI) for real-time fingerprint spoof detection which allows a visual examination of the local regions of the fingerprint highlighted as bonafide (live) or PA (spoof/altered).

- Optimized *Fingerprint Spoof Buster* by K-means ($K = 10$) clustering of minutiae followed by weighted fusion to reduce the required number of inferences (typically a $70\% - 80\%$ reduction.). Further, network architecture optimizations and quantization of model weights enabled development of a light-weight version of the proposed PAD, called *Fingerprint Spoof Buster Lite*[9], as an Android application which accepts a live-scan fingerprint and makes a bonafide vs. PA decision in 100ms on a commodity smartphone (Samsung Galaxy S8).

[8]Based on verbal communication

[9]We use the term *lite* to indicate a light version of the PAD as we utilize *TensorFlow Lite* framework for the proposed model optimizations. https://www.tensorflow.org/lite

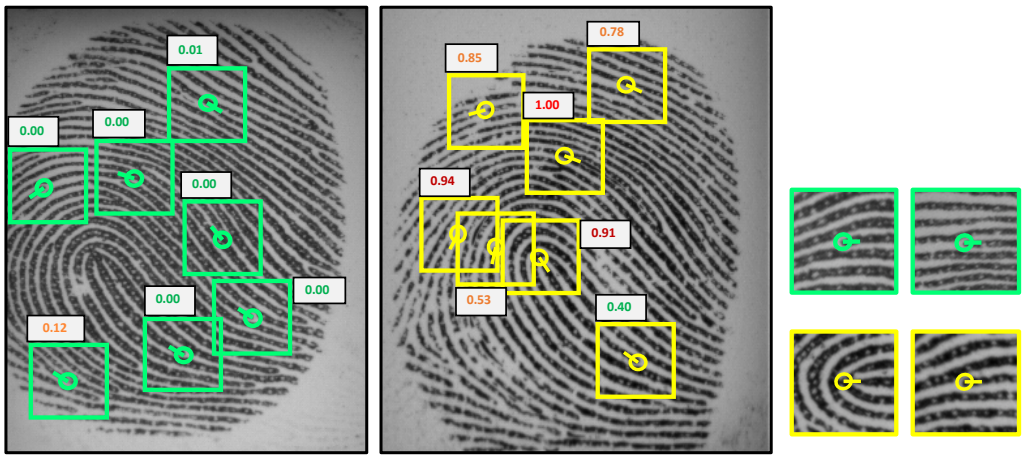


Figure 2.6 Local patches extracted around the fingerprint minutiae for (a) real fingerprint, and (b) spoof fingerprint (gelatin), and (c) aligned using minutiae orientation. The spoofness score for each patch is in the range $[0-1]$; higher the score, more likely the patch is extracted from a spoof fingerprint. For a given input test image, the spoofness scores corresponding to the local patches are averaged to give a global spoofness score. The final decision is made based on a classification threshold learned from the training dataset; an image with a global spoofness score below the threshold is classified as live, otherwise as spoof. Only a subset of detected fingerprint minutiae are shown for illustrative purposes.

Fingerprint Spoof Buster consists of two stages, an offline training stage and an online testing stage. The offline training stage involves (i) detecting minutiae in the sensed fingerprint image, (ii) extracting local patches centered and aligned using minutiae location and orientation, respectively, and (iii) training MobileNet models on the aligned local patches. During the testing stage, the spoof detection decision is made based on the average of spoofness scores for individual patches output from the MobileNet model. An overview of the proposed approach is presented in Fig. 2.5.

2.3.1 Minutiae Extraction

The fingerprint minutiae are extracted using the algorithm from [16]. The four LivDet datasets (LivDet 2011, 2013, 2015, and 2017) comprise of fingerprint images captured at varying resolutions, ranging from 500 dpi to 1000 dpi. Since the minutiae detector in [16] was designed for 500 dpi images, all fingerprint images are resized to ensure a standard resolution of 500 dpi. A standard

resolution for all the fingerprint images is also crucial to ensure similar amount of friction ridge area in each local patch, irrespective of the fingerprint reader used. An average of 46 minutiae (std. dev. = 6.2) and 50 minutiae (std. dev. = 6.9) are detected per live image and spoof image, respectively, for these LivDet datasets.

2.3.2 Local Patch Extraction

For a given fingerprint image I with k detected minutiae points $M = \{m_1, m_2, \ldots, m_k\}$, where $m_i = \{x_i, y_i, \theta_i\}$, *i.e.*, the minutiae m_i is defined in terms of spatial coordinates (x_i, y_i) and orientation (θ_i), a corresponding set of k local patches $L = \{l_1, l_2, \ldots, l_k\}$, each of size [$q \times q$] where ($q = \sqrt{2}p$), are extracted. Each local patch l_i, centered at the corresponding minutia location (x_i, y_i), is aligned[10] based on the minutiae orientation (θ_i). After alignment, the central region of size [$p \times p$] ($p = 96$) is cropped from the rotated patch and used for training the CNN model. The size of larger patch is fixed to [$\sqrt{2}p \times \sqrt{2}p$] to prevent any loss of information during patch alignment. Fig. 2.6 presents examples of real and spoof fingerprint images and the corresponding local patches centered and aligned using minutiae location and orientation, respectively.

For evaluating the impact of local patch size on the spoof detection performance, we also explore use of multi-resolution patches of size $p \in \{64, 96, 128\}$ for training independent CNN models and their fusion. All the local patches are resized[11] to 224×224 as required by the Mobilenet-v1 model.

2.3.3 MobileNet CNN

Since the success of AlexNet [93] for object detection in ILSVRC-2012 [140], different CNN architectures have been proposed in literature, such as VGG, GoogleNet (Inception v1-v4), ResNets, MobileNet, etc. Nogueira et al. [119], winner of LivDet 2015, utilized a pre-trained VGG ar-

[10]MATLAB's *imrotate* function with bilinear interpolation is used to rotate the local patch for alignment.

[11]TensorFlow's resize utility with bilinear interpolation was used; available at https://www.tensorflow.org/api_docs/python/tf/image/resize_images

chitecture [147] to achieve the best performance in LivDet 2015 [113]. In this study, we utilize the MobileNet-v1 architecture [71] because it offers the following advantages over other network architectures (such as VGG and Inception-v3): (i) MobileNet-v1 is designed using depth-wise separable convolutions, originally introduced in [21], providing drastic decrease in model size and training/evaluation times while providing better spoof detection performance; (ii) it is a low-latency network requiring only 6ms to classify an input fingerprint patch as live or spoof compared to 50ms required by Inception-v3 network [23] using a Nvidia 1080Ti GPU; and (iii) the number of model parameters to be trained in MobileNet-v1 (4.24M) is significantly smaller than the number of model parameters in Inception-v3 (23.2M) and VGG (138M), requiring significantly lower efforts in terms of regularization and data augmentation to prevent overfitting [71].

We utilized the TF-Slim library[12] implementation of the MobileNet-v1 architecture. The last layer of the architecture, a 1000-unit softmax layer (originally designed to predict the $1,000$ classes of ImageNet dataset), was replaced with a 2-unit softmax layer for the two-class problem, *i.e.*, live vs. spoof. The optimizer used to train the network was RMSProp with asynchronous gradient descent and a batch size of 100. Data augmentation techniques, such as brightness adjustment, random cropping, and vertical flipping, are employed to ensure the trained model is robust to the possible variations in fingerprint images. For the multi-resolution local patches, a separate network is trained for each patch size with the same parameters as mentioned above.

2.3.4 Fine-grained Fingerprint Image Representation

Partial spoofs and fingerprint alterations are meant to avoid re-identification[13], by masking the true identity from a fingerprint biometric system [23, 170]. Spoof detectors trained on the whole fingerprint images are ineffective against localizing partial spoof fingerprints, that conceal only a limited region of the live finger. Moreover, in many smartphones and other embedded systems that only sense a partial region (friction ridge area) of the fingerprint due to small sensor area (typically

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

[13] http://abcnews.go.com/Technology/GadgetGuide/surgically-altered-fingerprints-woman-evade-immigration/story?id=9302505

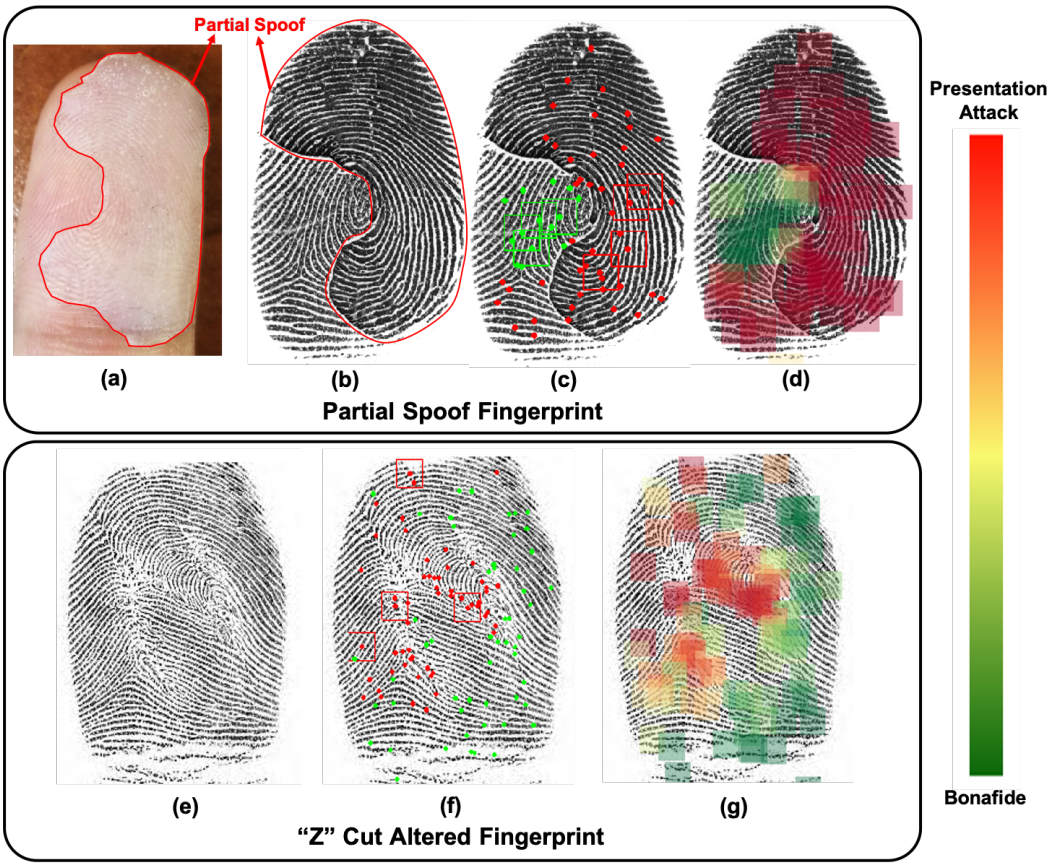


Figure 2.7 The proposed approach provides a fine-grained representation for spoof detection by using minutiae-based local patches. A fingerprint spoof fabricated using silicone which conceals only a partial region of the live finger is shown in (a) and the imaged fingerprint in (b) (enclosed in red). The proposed approach extracts and evaluates the minutiae-based local patches, and highlights the local regions as live (in green) or spoof (in red) as shown in (c) and (d). It can also highlight the regions of fingerprint alterations as shown for a "Z" cut altered fingerprint in (e), (f) and (g). The proposed approach detected (b) and (e) as spoofs with the spoofness scores of 0.78 and 0.65, respectively. (Best viewed in color)

150×150), it is very crucial to have a detailed representation of the sensed fingerprint region. One of the key advantages of employing a patch-based approach is the fine-grained representation of input fingerprint image for spoof detection. Fig. 2.7 (a) presents an example of a fingerprint spoof fabricated using silicone, concealing only a partial region of the live finger and Fig. 2.7 (b) presents the imaged partial spoof fingerprint using a CrossMatch Guardian 200 fingerprint reader. The proposed approach, utilizing minutiae-based local patches, highlights the local regions as live or spoof (shown in Figs. 2.7 (c) and (d) in green and red, respectively), providing a fine-grained representation of the fingerprint image. Fingerprint alterations, such as cuts, mutilations, stitches,

etc., performed using surgical and chemical procedures (see Fig. 2.7 (e)), create spurious minutiae as shown in Figs. 2.7 (f) and (g). The proposed approach is able to highlight the regions of fingerprint alterations despite not being trained specifically on altered fingerprint database, indicating the generalizability of the proposed approach. The proposed approach detected both fingerprint images in Figs. 2.7 (b) and (e) as spoofs with the spoofness scores of 0.78 and 0.65, respectively.

2.3.5 Spoofness Score

The output from the softmax layer of the trained MobileNet-v1 model is a spoof probability score, called as the *Spoofness Score*, in the range $[0, 1]$. The larger the spoofness score (close to 1), the higher the support that the input local patch belongs to the spoof class (see Fig. 2.6). For an input test image I, the spoofness scores $s^I_{i\in\{1,2,\ldots,k\}}$ corresponding to the k minutiae-based local patches of size $p \times p$, extracted from the input image, are averaged to give a global spoofness score S^I. In case of multi-resolution local patches, the global spoofness scores ($S^I_{p_i}$) based on each local patch size, $p_i \in \{64, 96, 128\}$, are averaged to produce a final spoofness score. The threshold that minimizes the average classification error on training dataset is learned and utilized as the classification threshold. An image with a spoofness score below the threshold is classified as live, otherwise as spoof. The learned threshold performed slightly better in spoof detection than selecting a pre-defined threshold of 0.5.

2.3.6 On Robustness of Patch-based Representation

While the proposed approach is based on the premise that it is capable of capturing discriminatory information from local patches (presence of artifacts), such as valley noise, broken ridges, air bubbles, etc., from spoof fingerprints, we also examine the robustness of patch-based representation by evaluating it in the absence of such artifacts. Figs. 2.8 (a) and (b) present minutiae-based local patches from a live fingerprint and the corresponding spoof fingerprint (fabricated using EcoFlex), respectively, for the same minutia point, and Figs. 2.8 (d) and (e) present their feature representations, respectively, obtained from the bottleneck layer of the MobileNet-v1 architecture. The

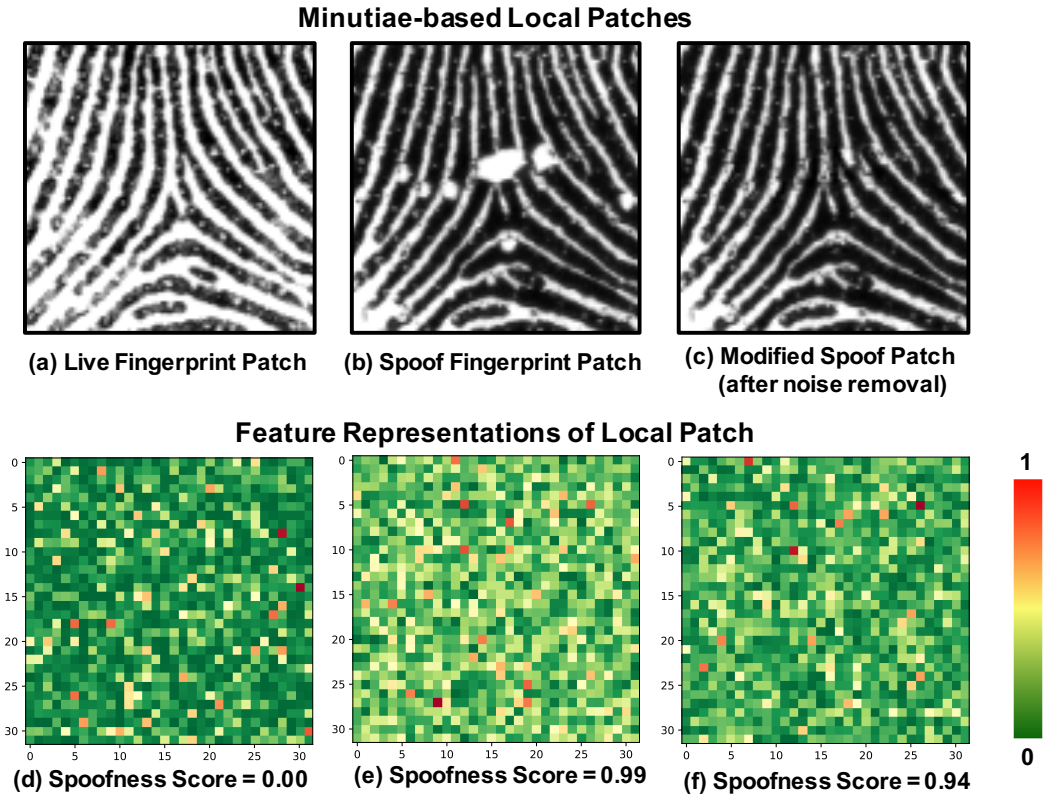


Figure 2.8 Illustrating the embeddings of minutiae-based local patches (96×96), for (a) live patch, (b) spoof patch, and (c) modified spoof patch (retouched to remove visible artifacts), in 1024-dimensional feature space from MobileNet-v1 bottleneck layer, transformed to 32×32 heat maps, (d), (e), and (f), respectively, for visualization. A high spoofness score for the modified spoof patch is achieved, despite removal of artifacts, indicating the robustness of the proposed approach. (Best viewed in color)

1024-dimensional feature representation is transformed to 32×32 heatmap for visualization. The spoofness scores for the two patches, live and spoof, are 0.00 (Fig. 2.8 (b)) and 0.99 (Fig. 2.8 (d)), respectively. The spoof patch (Fig. 2.8 (b)) is modified, by the authors, using an open-source photo-editing utility, called *GIMP*[14], to remove the visible artifacts and produce the modified spoof fingerprint patch as shown in Fig. 2.8 (c). The feature representation for the modified patch is shown in Fig. 2.8 (f). A high spoofness score for the modified spoof patch (0.94) despite removal of artifacts indicates the robustness of the proposed approach.

[14]https://www.gimp.org/

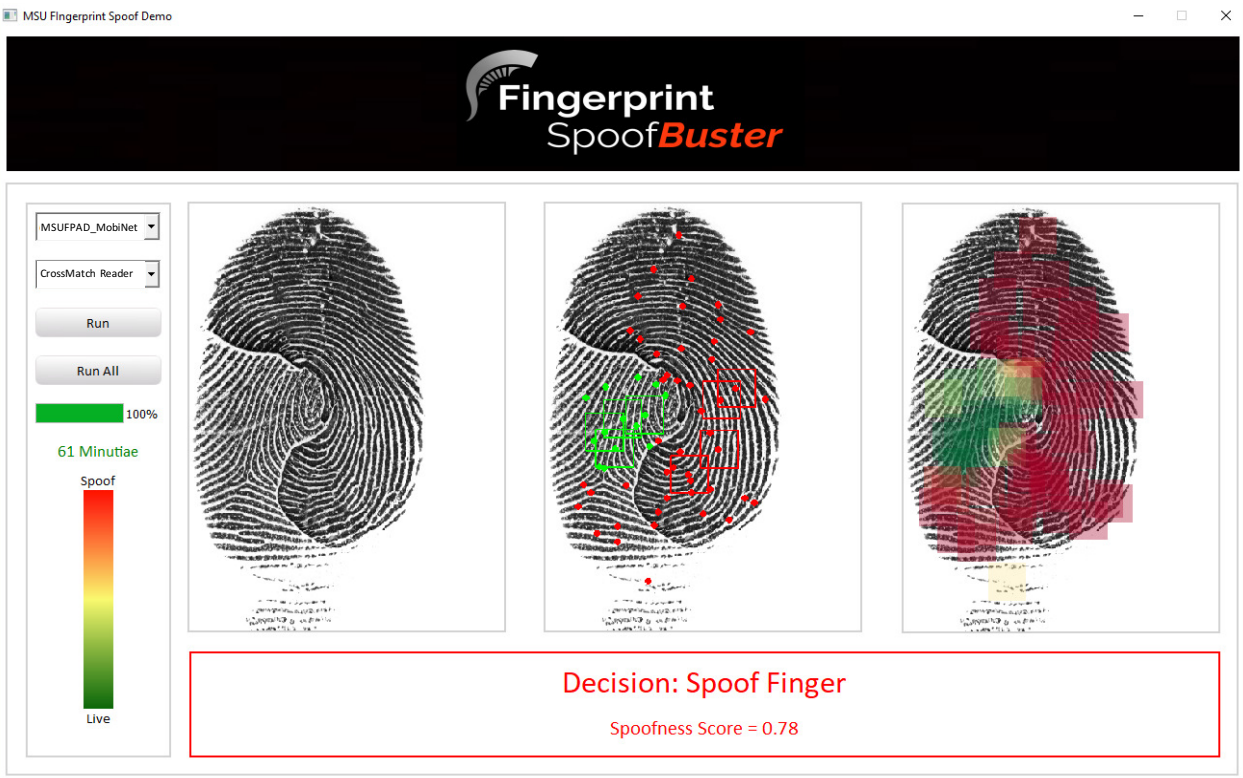


Figure 2.9 Interface of the proposed Fingerprint Spoof Buster. It allows selection of the fingerprint reader and CNN model. (Best viewed in color)

2.3.7 Graphical User Interface (GUI)

A graphical user interface for Fingerprint Spoof Buster allows the operator to select a specific fingerprint reader and a trained MobileNet-v1 model for evaluation. The operator can perform the evaluation in either *online* or *batch* mode. In the *online* mode, a fingerprint is imaged using the selected reader and displayed on the interface (see Fig. 2.9). The extracted fingerprint minutiae and the corresponding local patches are presented and color-coded based on their respective spoofness scores (green for live and red for spoof). The global spoofness score and the final decision for the input image is also presented on the interface. In the batch mode, all fingerprint images within a specified directory are evaluated, and global spoofness scores for each fingerprint file are output together in a score file. The graphical user interface allows the operator to visually examine the local regions of the fingerprint highlighted as live or spoof, instead of relying on only a single score as output by the traditional approaches.

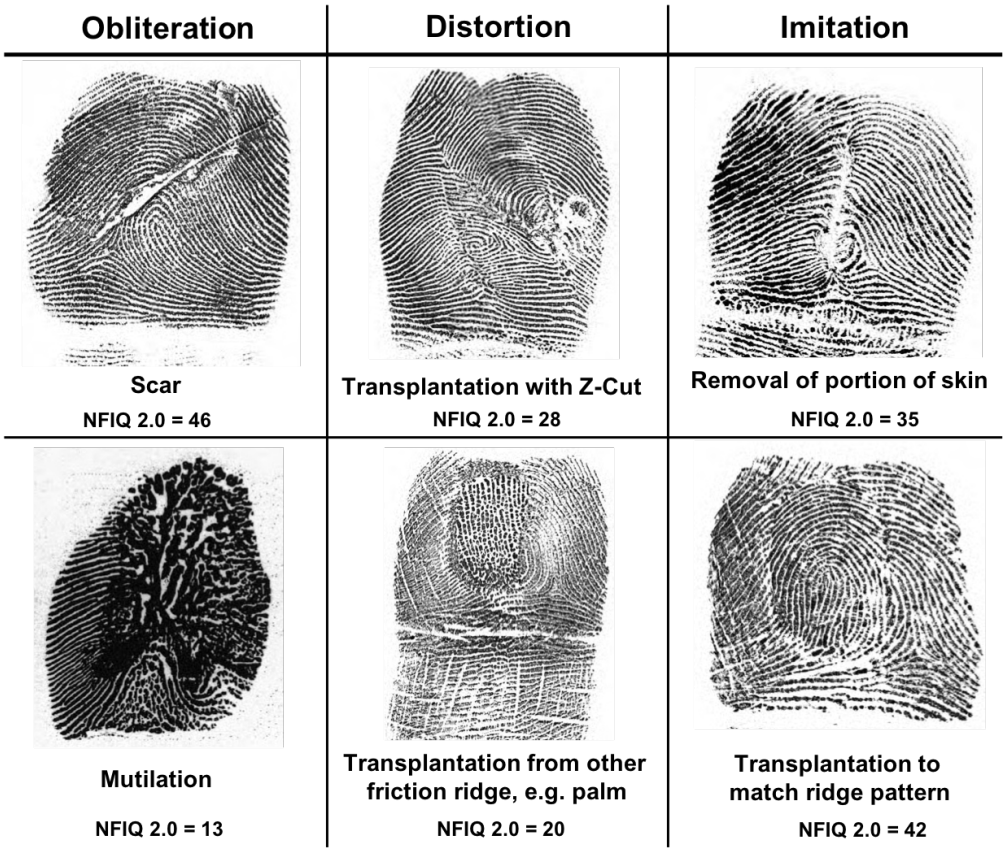


Figure 2.10 Types of fingerprint alterations: (i) Obliteration, such as scars, or mutilations, (ii) Distortion, *i.e.*, friction ridge transplantation to distort friction ridge area, and (iii) Imitation, *i.e.*, transplantation or removal of friction ridge skin while still preserving fingerprint like pattern.

2.4 Altered Fingerprints: Detection and Localization

2.4.1 Altered Fingerprint Detection

The goal of detecting altered fingerprint images can be formulated as a binary classification problem with two classes; *altered* and *bonafide*. While some cuts and cruises could be due to unintentional accidents, our interest here is to detect any fingerprint where the bonafide ridge structure is significantly modified. As shown in Figure 2.10, different types of alteration procedures would result in different fingerprint degradation. Different types of alteration procedures and their effect on friction ridge patterns are discussed in [40, 170]. Based on the changes made to friction ridge patterns, they categorized altered fingerprints into three types: *obliteration*, *distortion*, and *imitation*.

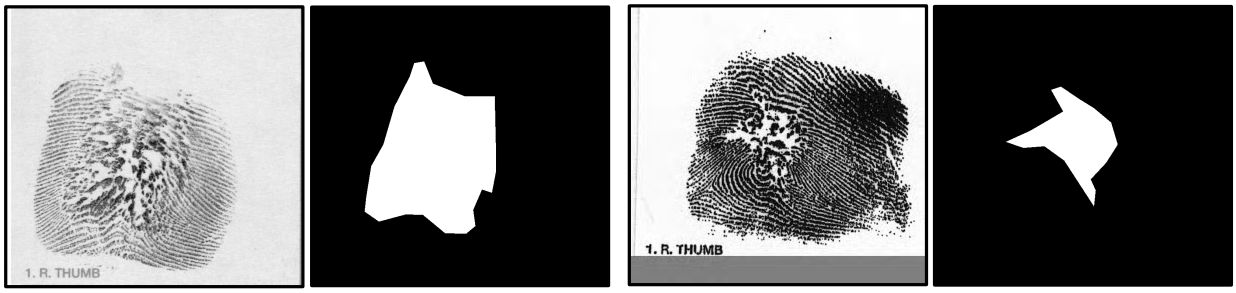


Figure 2.11 Examples of altered fingerprints and corresponding manually marked regions of interest (ROI) circumscribing the areas of fingerprint alterations. Local patches overlapping with manually marked ROI are labeled as altered patches, while the rest are labelled as bonafide. The test phase is fully automatic and does not require any manual markup.

Obliteration consists of abrading, cutting, burning, applying strong chemicals, or transplanting friction ridge skin. Skin disease or side effects of drugs can also obliterate fingertips. *Distortion* comprises of cases of using plastic surgery to convert a normal friction ridge pattern into an unusual ridge pattern. Some portions of skin are removed from the finger and grafted back onto a different position causing an unusual pattern. *Imitation* is when a surgical procedure is performed in such a way that the altered fingerprints appear as natural fingerprints, for example, by grafting skin from the other hand or a toe such that fingerprint ridge pattern is still preserved. Despite Yoon and Jain's [170] suggestion to develop different models for different alteration types, we propose to utilize a single model for the following two reasons: a) insufficient data for each alteration type for training deep networks, and b) manual labeling of the alteration type would be subjective because an image can suffer from more than one alteration type. We trained a Convolutional Neural Network (CNN) to classify an input fingerprint image into one of the two classes of bonafide or altered. Data augmentation techniques, such as mirroring, random cropping, and rotation have been employed to increase the size of the training data.

2.4.2 Localization of Altered Regions

To localize and highlight the altered regions of fingerprints, we augment our whole image based altered fingerprint detection with a patch-based approach. Our approach is as follows: First, re-

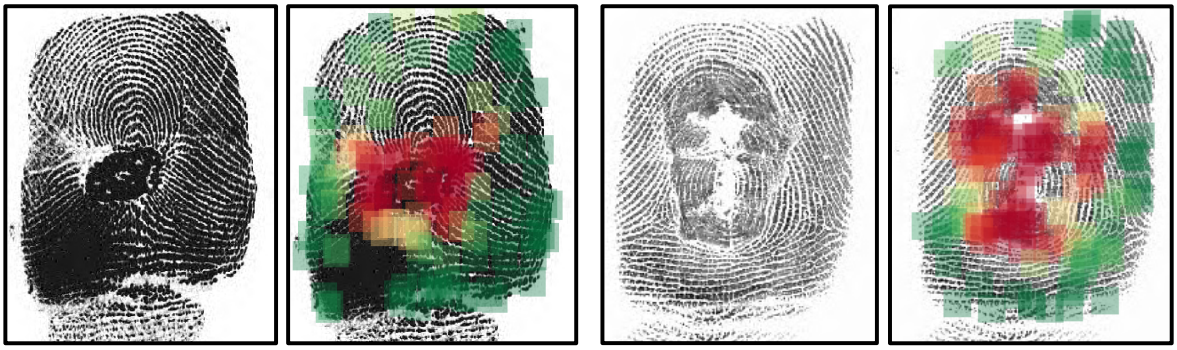


Figure 2.12 Examples of altered fingerprint localization by our proposed method. Local regions highlighted in red represent the altered portion of the fingerprint, whereas regions highlighted in green reflect the bonafide friction ridge area. (Best viewed in color)

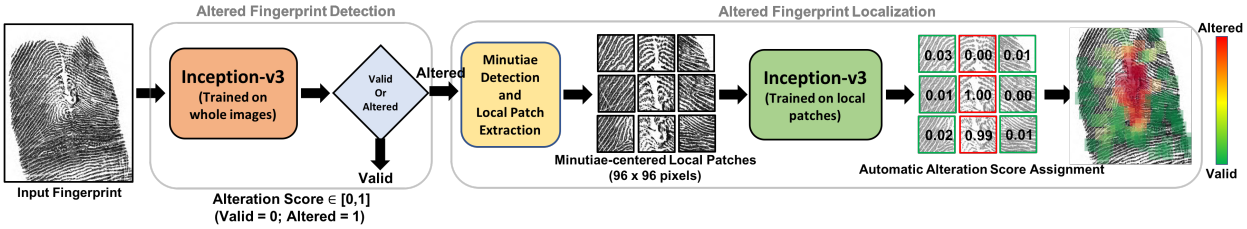


Figure 2.13 An overview of the proposed approach for detection and localization of altered fingerprints. We trained two convolutional neural networks (Inception-v3 and Mobilenet-v1) using full fingerprint images and local patches of images where patches are centered on minutiae locations.

gion of interest (ROI) is manually marked for $1,182$ randomly selected altered fingerprints from our database of $4,815$ altered fingerprints. See Figure 2.11. Next, local patches of size 96×96 centered around each extracted minutia are cropped. Local patches with more than 50% overlap with the manually marked ROI are labeled as altered patches, and the remaining patches are labeled as bonafide. Because a majority of fingerprint alterations generate discontinuities and noisy regions in the friction ridge pattern, a much higher number of spurious minutiae are generated in altered fingerprints compared to bonafide fingerprints of the same size [170]. As discussed earlier, local patches centered around minutiae provide superior performance in fingerprint spoof detection compared to patches extracted in a raster scan or random manner. A total of $81,969$ bonafide and $89,979$ altered patches are extracted and utilized for training two different networks: Inception-v3 [150] and MobileNet-v1 [71]. Fig. 2.12 presents examples of altered fingerprint localization

Table 2.3 Network hyper-parameters utilized in training CNN models for altered fingerprint detection and localization.

Hyper-paramters	Inception-v3	MobileNet-v1
Batch Size	32	100
Optimizer	RMSProp	RMSProp
Learning Rate	[0.01 - 0.0001]; exp. decay 0.94	[0.01 - 0.0001]; exp. decay 0.94
Momentum	0.9	0.9
Iterations	75,000	25,000

output by the proposed approach. An overview of the proposed approach to detect and localize altered fingerprints is presented in Figure 2.13.

2.4.3 Alteration Score

We train MobileNet-v1 [71] and Inception-v3 [150] networks, using TF-Slim library [145], as binary classifiers (altered vs. bonafide fingerprints). The input is a full fingerprint image and the output is a probability (or score) of belonging to Altered or Valid class, referred to as *alteration score*. A bonafide fingerprint image should result in an alteration score of close to 0, whereas an altered fingerprint image should result in an alteration score of close to 1. The network hyper-parameters used to train the CNN models are presented in Table 2.3.

2.5 End-to-End Presentation Attack Detection

The proposed modules for altered fingerprint detection and spoof detection can be implemented in a cascaded manner as shown in Figure 2.14. First, a whole fingerprint image is fed to the altered fingerprint detector. If the input image is classified as an altered fingerprint, we output the alteration score and evaluate minutiae-based local patches to localize the altered regions. Otherwise, if the image is classified as valid, it is then fed to the Fingerprint Spoof Buster for spoof detection, which evaluates the whole image and minutiae-based local patches and performs average score fusion to generate a global spoofness score. It also outputs a heat map overlaid on the input image

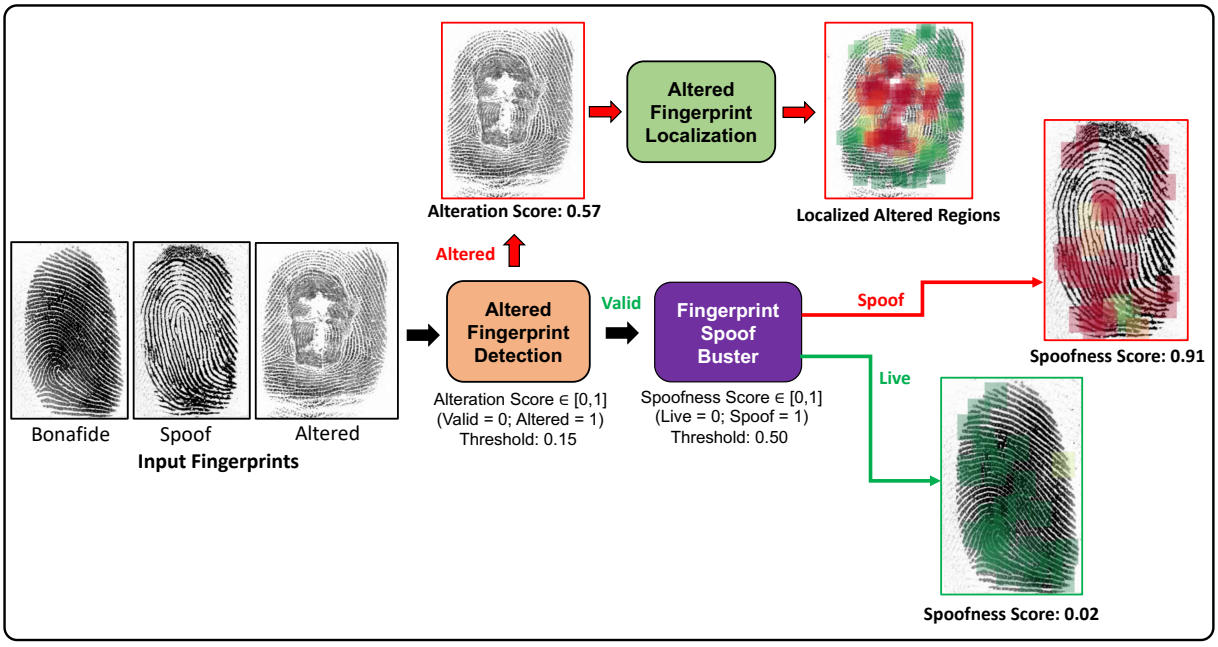


Figure 2.14 An overview of the proposed end-to-end presentation attack detection. (Best viewed in color)

highlighting the spoof and bonafide regions. The score thresholds for altered fingerprint detection and spoof detection are set to 0.15 and 0.50, respectively.

2.6 Experimental Results

2.6.1 Performance Evaluation Metrics

The performance of the proposed approach is evaluated following the metrics used in LivDet [57].

- $Errlive$: Percentage of misclassified live fingerprints.
- $Errfake$[15]: Percentage of misclassified spoof fingerprints.

[15]When all the spoof fabrication materials are known during the training, this metric is referred to as $Errfake\_known$, and in case all the spoof fabrication materials to be encountered during testing are not known during training, this metric is referred to as $Errfake\_unknown$.

Table 2.4 Summary of the Liveness Detection (LivDet) datasets (LivDet 2011 and LivDet 2013) utilized in this study.

Dataset	**LivDet 2011 [167]**				**LivDet 2013 [58]**	
Fingerprint Reader	Biometrika	ItalData	Digital Persona	Sagem	Biometrika	ItalData
Model	FX2000	ET10	4000B	MSO300	FX2000	ET10
Image Size	315×372	640×480	355×391	352×384	315×372	640×480
Resolution (dpi)	500	500	500	500	569	500
#Live Images Train / Test	1000/1000	1000/1000	1000/1000	1000/1000	1000/1000	1000/1000
#Spoof Images Train / Test	1000/1000	1000/1000	1000/1000	1000/1000	1000/1000	1000/1000
Cooperative Subject	Yes	Yes	Yes	Yes	No	No
Spoof Materials	Ecoflex, Gelatine, Latex, Silgum, Wood Glue		Gelatine, Latex, Play Doh, Silicone, Wood Glue		Ecoflex, Gelatine, Latex, Modasil, Wood Glue	

The average classification error (ACE) is defined as:

$$ACE = \frac{F_{errlive} + F_{errfake}}{2} \tag{2.6.1}$$

Additionally, we also report the $Ferrfake$ @ $Ferrlive = 1.0\%$ for each of the experiments as reported in [57]. This value represents the percentage of spoofs able to breach the biometric system security when the reject rate of legitimate users $\leq 1.0\%$.

2.6.2 Presentation Attack Datasets

The following datasets have been utilized to evaluate the proposed approach:

2.6.2.1 LivDet Datasets

In order to evaluate performance of the proposed approach, we utilized LivDet 2011 [167], LivDet 2013 [58], LivDet 2015 [113] and LivDet 2017 [114] datasets. Each of these datasets contains over $16,000$ fingerprint images, acquired from three or more different fingerprint readers, with comparable numbers of live and spoof fingerprints. However, the CrossMatch and Swipe readers from LivDet 2013 dataset were not utilized for evaluation purposes because the (a) LivDet compe-

Table 2.5 Summary of the Liveness Detection (LivDet) datasets (LivDet 2015 and LivDet 2017) utilized in this study.

Dataset	**LivDet 2015 [113]**				**LivDet 2017 [114]**		
Fingerprint Reader	GreenBit	Biometrika	Digital Persona	CrossMatch	GreenBit	Orcanthus	Digital Persona
Model	Dacty Scan26	HiScan-PRO	U.are.U 5160	L Scan Guardian	Dacty Scan 84C	Certis2 Image	U.are.U 5160
Image Size	500×500	1000×1000	252×324	800×750	500×500	$300 \times n$	252×324
Resolution (dpi)	500	1000	500	500	569	500	500
#Live Images Train / Test	1000/1000	1000/1000	1000/1000	1510/1500	1000/1700	1000/1700	999/1692
#Spoof Images Train / Test	1000/1500	1000/1500	1000/1500	1473/1448	1200/2040	1200/2018	1199/2028
Cooperative Subject	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Spoof Materials	Ecoflex, Gelatine, Latex, Wood Glue, Liquid Ecoflex, RTV			Body Double, Ecoflex, Play Doh, OOMOO, Gelatin	Wood Glue, Ecoflex, Body Double, Gelatine, Latex, Liquid Ecoflex		

tition organizers found anomalies in the fingerprint data from CrossMatch reader and discouraged its use for comparative evaluations [57], and (b) the resolution of fingerprint images output from Swipe reader is very low, *i.e.*, 96 dpi. Unlike other LivDet datasets, spoof fingerprint images from Biometrika and Italdata readers in LivDet 2013 dataset [58] are fabricated using the *non-cooperative method*, *i.e.*, without user cooperation. It should be noted that in LivDet 2015 and LivDet 2017, the testing set included spoofs fabricated using new materials, that were not known in the training set. In the case of LivDet 2015, these new materials included liquid ecoflex and RTV for Biometrika, Digital Persona, and Green Bit readers, and OOMOO and gelatin for Crossmatch reader. In the case of LivDet 2017, the testing set contained materials, namely Gelatine, Latex, and Liquid Ecoflex, completely different from training which contained Wood Glue, Ecoflex, and Body Double materials. Tables 2.4 and 2.5 presents a summary of the LivDet datasets used in this study.

2.6.2.2 MSU Fingerprint Presentation Attack Dataset

In addition to utilizing LivDet Datasets, we collected a large dataset, called the MSU Fingerprint Presentation Attack Dataset (MSU-FPAD), using two different fingerprint readers, namely, Cross-Match Guardian 200 and Lumidigm Venus 302. There are a total of $9,000$ live images and $10,500$

Table 2.6 Summary of the MSU Fingerprint Presentation Attack Dataset (MSU-FPAD) and Precise Biometrics Spoof-Kit Dataset (PBSKD).

Dataset	**MSU-FPAD**		**Precise Biometrics Spoof-Kit**	
Fingerprint Reader	CrossMatch	Lumidigm	CrossMatch	Lumidigm
Model	Guardian 200	Venus 302	Guardian 200	Venus 302
Image Size	750×800	400×272	750×800	400×272
Resolution (dpi)	500	500	500	500
#Live Images Train / Test	$2,250$ / $2,250$	$2,250$ / $2,250$	250 / 250†	250 / 250†
#Spoof Images Train / Test	$3,000$ / $3,000$	$2,250$ / $2,250$	250 / 250	200 / 200‡
Cooperative*	Yes	Yes	Yes	Yes
Spoof Materials	Ecoflex, PlayDoh, 2D Print (Matte Paper), 2D Print (Transparency)		Ecoflex, Gelatin, Latex body paint, Ecoflex with silver colloidal ink coating, Ecoflex with BarePaint coating, Ecoflex with Nanotips coating, Crayola Model Magic, Wood glue, Monster Liquid Latex, and 2D printed fingerprint on office paper	

†1000 randomly sampled live fingerprint images from MSU-FPAD are selected for Precise Biometrics Spoof-Kit Dataset.

‡ Lumidigm fingerprint reader does not image Silicone (EcoFlex) spoofs with NanoTips and BarePaint coatings.

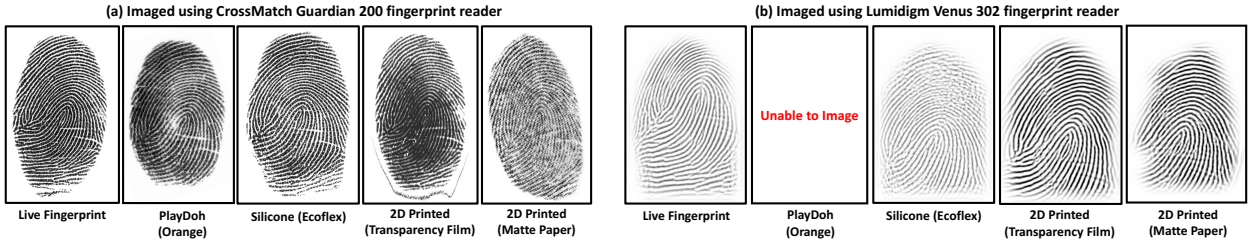


Figure 2.15 Example images from MSU Fingerprint Presentation Attack Dataset (MSU-FPAD) acquired using (a) CrossMatch Guardian 200, and (b) Lumidigm Venus 302 fingerprint readers. Note that Lumidigm reader does not image PlayDoh (orange) spoofs.

spoof images captured using these two readers and 4 different spoof fabrication materials, namely, ecoflex, PlayDoh, 2D printed on matte paper, and 2D printed on transparency film. The selection of the fingerprint readers and the spoof materials is based on the requirements of IARPA ODIN program [123] evaluation. Fig. 2.15 presents some example fingerprint images, and Table 2.6 presents a summary of the MSU Fingerprint Presentation Attack Dataset.

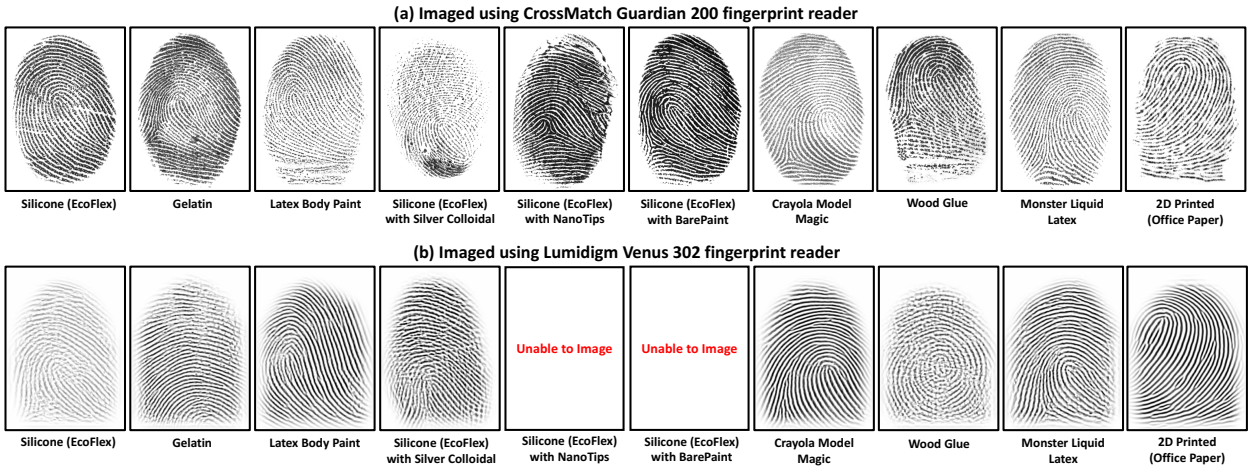


Figure 2.16 Example images from Precise Biometrics Spoof-Kit Dataset (PBSKD) acquired using (a) CrossMatch Guardian 200, and (b) Lumidigm Venus 302 fingerprint readers. Note that Lumidigm reader does not image Silicone (EcoFlex) spoofs with NanoTips and BarePaint coatings.

2.6.2.3 Precise Biometrics Spoof-Kit Dataset

We also collected another dataset containing 900 high quality spoof fingerprint images fabricated using 10 different types of spoof materials, namely, (i) Ecoflex, (ii) Gelatin, (iii) Latex body paint, (iv) Ecoflex with silver colloidal ink coating, (v) Ecoflex with BarePaint coating, (vi) Ecoflex with Nanotips coating, (vii) Crayola Model Magic, (viii) Wood glue, (ix) Monster Liquid Latex, and (x) 2D printed fingerprint on office paper. The spoof specimens used for this dataset are taken from Precise Biometrics[16] Spoof-Kit containing 10 specimens per spoof type, for a total of 100 spoof specimens. Each spoof specimen is imaged 5 times using two fingerprint readers, namely, CrossMatch Guardian 200 and Lumidigm Venus 302. Note that Lumidigm reader does not image Silicone (EcoFlex) spoofs with NanoTips and BarePaint coatings. An additional 900 randomly sampled live fingerprints from MSU-FPAD are selected for a total of $1,800$ fingerprint images in Precise Biometrics Spoof-Kit Dataset. Fig. 2.16 presents some example fingerprint images, and Table 2.6 presents a summary of the Precise Biometrics Spoof-Kit Dataset.

[16]https://precisebiometrics.com/

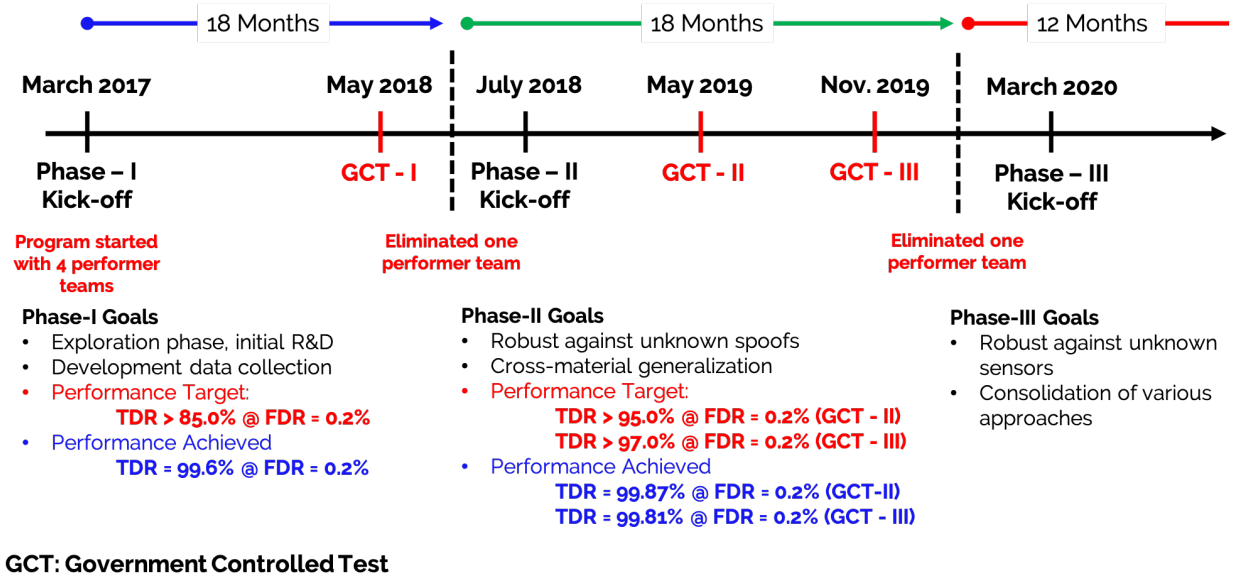


Figure 2.17 Illustration of the timeline of IARPA ODIN Program [123]. The Phase-III will be completed in March 2021.

2.6.2.4 Government Evaluation Datasets (GCT - I, II, and III)

During May 14 - May 25, 2018, the first Government Controlled Test- I (GCT-I), as part of the IARPA ODIN program [123], was organized. A total of $13,062$ fingerprint images were collected using two optical readers, CrossMatch Guardian 200 and Lumidigm V302, from 340 subjects in a span of 2 weeks at Johns Hopkins University Applied Physics Lab (JHU APL), Laurel, MD. Subjects presented either bonafide fingerprints or presentation attacks for a total of 20 impressions per sensor per subject. Four different PA types were used, namely, Transparency, Dragon Skin, Yellow Pigmented Silicone, and VeroBlack plus.

In the following year, during May 8 - May 17, 2019, Government Controlled Test - II (GCT-II) was conducted at JHU facility in Columbia, Maryland. A total of $8,598$ fingerprint images from around 400 subjects were collected on CrossMatch Guardian 200, including $7,852$ bonafide and 746 PA images fabricated with more than 8 PA types. Eight PA types were known (*i.e.*, seen by Spoof Buster during training), namely, Ballistic Gelatin, Clear Ecoflex, Tan Ecoflex, Yellow Pigmented Silicone, Flesh Pigmented Ecoflex, Nusil R-2631 Conductive Silicone, Flesh Pigmented

Table 2.7 Summary of the datasets collected during Government Controlled Test (GCT) I, II, and III as part of the IARPA ODIN program [123].

	GCT-I		GCT-II	GCT-III
	CrossMatch	**Lumidigm**	**CrossMatch**	**CrossMatch**
# Subjects	340	340	400	685
# PA Types	4	4	8+	12
# Bonafide Samples	6,781	5,842	7,852	13,241
# PA Samples	232	207	746	1,049
Total	7,013	6,049	8,598	14,290

PDMS, and Elmer's Glue. A few fingerprint presentations obfuscated with bandaids were also labeled as PA.

More recently, during Oct. 28 - Nov. 15, 2019, Government Controlled Test - III (GCT-III) was conducted at JHU facility in Columbia, Maryland. A total of $14,290$ fingerprint images from 685 subjects were collected on CrossMatch Guardian 200, including $13,241$ bonafide and $1,049$ PA images fabricated with more than 12 PA types. Figure 2.17 presents the timeline of the IARPA ODIN Program. Table 2.7 summarizes the number of bonafide and PA samples collected in the three government evaluation datasets.

2.6.2.5 Altered Fingerprint Dataset

An operational dataset of $4,815$ altered fingerprints, from 635 tenprint cards of 270 subjects [170], acquired from law enforcement agencies is utilized to evaluate the proposed approach. The number of tenprint cards per subject varies from 1 to 16 due to multiple encounters. However, not all 10 fingerprint images in a tenprint card may be altered. The number of altered fingerprint instances per subject varies from 1 to 137. Another operational dataset of $4,815$ rolled fingerprint images is used for bonafide fingerprints [15]. Fingerprint images in both sets of altered and bonafide are images collected as part of law enforcement operations. All images are 8-bits gray scale. Figure 2.18 shows distribution of NFIQ 2.0 [75, 151] scores for the altered and bonafide fingerprint images used in this study[17]. A five-fold cross-validation is employed where in each of the five folds, the training set contains $3,852$ altered and $3,852$ bonafide fingerprints. The testing set in each fold

[17]NFIQ 2.0 software reads a fingerprint image, computes a set of quality features from the image, and uses these features to predict the utility of the image as an integer score between 0 and 100.

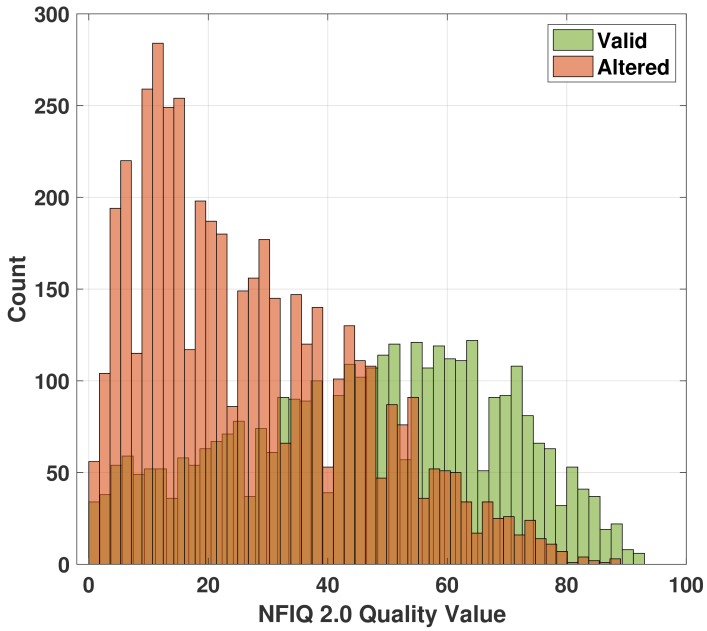


Figure 2.18 Histogram of NFIQ 2.0 quality scores for bonafide/valid (green) and altered (red) fingerprint images. Approximately, 75% of altered fingerprint images have a NFIQ 2.0 score of 40 or lower, and only 10% of altered dataset has a NFIQ 2.0 score of larger than 50. The median NFIQ 2.0 score for altered fingerprint images is 23, while median NFIQ 2.0 score for bonafide fingerprint images is 48. This suggests NFIQ 2.0's suitability for detecting altered fingerprints, particularly for cases of fingerprint obliteration. (Best viewed in color)

contains the remaining 963 altered and 963 bonafide fingerprints, such that the train and test sets are disjoint. Figure 2.19 shows sample altered and bonafide images used for training and testing in one of the five folds.

2.6.3 Spoof Detection Results

The proposed approach is evaluated under the following four scenarios of fingerprint spoof detection, which reflect an algorithm's robustness against new spoof materials, use of different sensors and/or different environments.

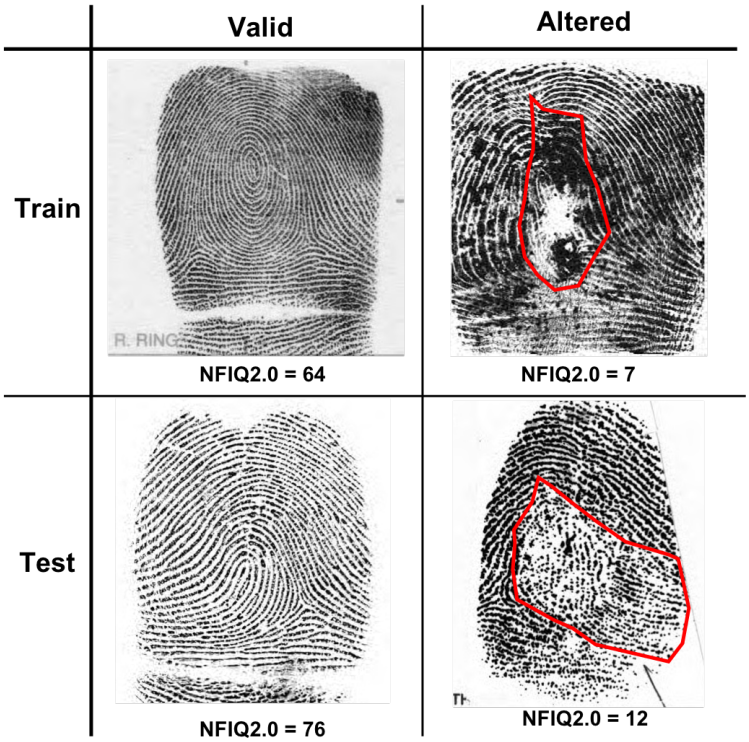


Figure 2.19 Example of altered and bonafide fingerprint images used for training and testing in one of the five folds. The altered region is highlighted in red. The NFIQ 2.0 quality scores are also presented for each image; the larger NFIQ 2.0 score, the higher fingerprint quality. The NFIQ 2.0 quality scores ranges between [0, 100].

2.6.3.1 Intra-Sensor, Known Spoof Materials

In this setting, all the training and testing images are captured using the same sensor, and all spoof fabrication materials utilized in the test set are known a priori. Our experimental results show that training the MobileNet-v1 model from scratch, using minutiae-based local patches, performs better than fine-tuning a pre-trained network, as reported in [119]. The large amount of available data, in the form of local fingerprint patches, is sufficient to train the deep architecture of MobileNet-v1 model without over-fitting.

It was reported in [57] that most of the algorithms submitted to LivDet 2015 did not perform well on Digital Persona sensor due to the small image size. Our approach based on local patches does not suffer from this limitation. Tables 2.8 and 2.9 present the performance comparison between the proposed approach and the state-of-the-art results for the LivDet datasets utilized in

Table 2.8 Performance comparison between the proposed approach (bottom) and state-of-the-art (top) reported on LivDet 2015 dataset [113]. Separate networks are trained on the training images captured by each of the four fingerprint readers. *Ferrfake known* and *Ferrfake unknown* correspond to Known Spoof Materials and Cross-Material scenarios, respectively.

	LivDet 2015	**Ferrlive (%)**	**Ferrfake† (%)**	**Ferrfake known (%)**	**Ferrfake unknown* (%)**	**ACE (%)**	**Ferrfake (%) @ Ferrlive= 1% [57]**
State-of-the-Art [113]	GreenBit	3.50	5.33	4.30	7.40	4.60	17.90
	Biometrika	8.50	3.73	2.70	5.80	5.64	15.20
	Digital Persona	8.10	5.07	4.60	6.00	6.28	19.10
	Crossmatch	0.93	2.90	2.12	4.02	1.90	2.66
	Average	4.78	4.27	3.48	5.72	4.49	13.24
Proposed Approach	**LivDet 2015**	**Ferrlive (%)**	**Ferrfake† (%)**	**Ferrfake known (%)**	**Ferrfake unknown* (%)**	**ACE (%)**	**Ferrfake (%) @ Ferrlive = 1%**
	GreenBit	0.50	0.80	0.30	1.80	**0.68**	0.53
	Biometrika	0.90	1.27	0.60	2.60	**1.12**	1.20
	Digital Persona	1.97	1.17	0.85	1.80	**1.48**	1.96
	Crossmatch	0.80	0.48	0.82	0.00	**0.64**	0.28
	Average	**1.02**	**0.93**	**0.64**	**1.48**	**0.97**	**0.96**

† Ferrfake includes spoofs fabricated using both known and previously unseen materials. It is an average of Ferrfake-known and Ferrfake-unknown, weighted by the number of samples in each category.
*The unknown spoof materials in LivDet 2015 test dataset include Liquid Ecoflex and RTV for Green Bit, Biometrika, and Digital Persona sensors, and OOMOO and Gelatin for Crossmatch sensor.

this study. Table 2.10 presents the performance of the proposed approach on MSU Fingerprint Presentation Attack Dataset (MSU-FPAD) and Precise Biometrics Spoof-Kit Dataset (PBSKD). Independent MobileNet-v1 networks are trained for each evaluation. Note that in LivDet 2015 (Table 2.8), this scenario is represented by the *Ferrfake known*. For LivDet 2011 and 2013, MSU-FPAD, and PBSKD datasets (Table 2.9), all spoof materials in the test set were known during training. Fig. 2.20 presents example fingerprint images for Biometrika sensor from LivDet 2015 dataset that were correctly and incorrectly classified by the proposed approach.

We also evaluate the impact of local patch size on the performance of the proposed approach, by comparing the performance of three CNN models trained on minutiae-centered local patches of size $[p \times p]$ where $p = \{64, 96, 128\}$, extracted from the fingerprint images captured by Biometrika sensor for LivDet 2011 dataset. Among these three models, the one trained on local patches of size $[96 \times 96]$ performed the best. However, a score-level fusion, using average-rule, of the three models reduced the average classification error (ACE) from 1.24% to 0.88%, and Ferrfake from 1.41% to

Table 2.9 Performance comparison between the proposed approach and state-of-the-art results reported on LivDet 2011 and LivDet 2013 datasets for intra-sensor experiments in terms of Average Classification Error (ACE) and Ferrfake @ Ferrlive = 1%.

Dataset	**State-of-the-Art**	**Proposed Approach**	
LivDet 2011	**ACE (%)**	**ACE (%)**	**Ferrfake @ Ferrlive = 1%**
Biometrika	4.90 [61]	**1.24**	1.41
Digital Persona	1.85 [126]	**1.61**	3.25
ItalData	5.10 [126]	**2.45**	7.21
Sagem	**1.23** [126]	1.39	4.33
Average	3.27	**1.67**	4.05
LivDet 2013			
Biometrika	0.65 [126]	**0.20**	0.00
ItalData	0.40 [119]	**0.30**	0.10
Average	0.53	**0.25**	0.05

Table 2.10 Average Classification Error (ACE), Ferrfake @ Ferrlive = 0.1% and Ferrlive = 1% on the MSU Fingerprint Presentation Attack Dataset (MSU-FPAD) and Precise Biometrics Spoof-Kit Dataset (PBSKD) for intra-sensor experiments.

Dataset	**Proposed Approach**		
MSU-FPAD	**ACE (%)**	**Ferrfake @ Ferrlive = 0.1%**	**Ferrfake @ Ferrlive = 1%**
CrossMatch Guardian 200	0.08	0.11	0.00
Lumidigm Venus 302	3.94	10.03	1.30
Average	**2.01**	**5.07**	**0.65**
PBSKD			
CrossMatch Guardian 200	2.02	5.32	0.65
Lumidigm Venus 302	1.93	3.84	0.33
Average	**1.98**	**4.66**	**0.51**

0.58% @ Ferrlive $= 1\%$. Similar performance gains were observed for other sensors, but there is a trade off between the performance gain and the computational requirements for the spoof detector. In order to evaluate the significance of utilizing minutiae locations for extracting local patches, we trained independent MobileNet-v1 models on a similar number of local patches, extracted randomly from LivDet 2015 datasets. It was observed that the models trained on minutiae-centered local patches achieved a significantly higher reduction (78%) in average classification error, compared to the reduction (33%) achieved by the models trained on randomly sampled local patches. Fig. 2.21 illustrates that (i) features extracted from local patches provide better spoof detection accuracy than the whole image, (ii) patches selected around minutiae perform better than random patches of the same size, (iii) 96×96 patch performs the best among the three patch sizes con-

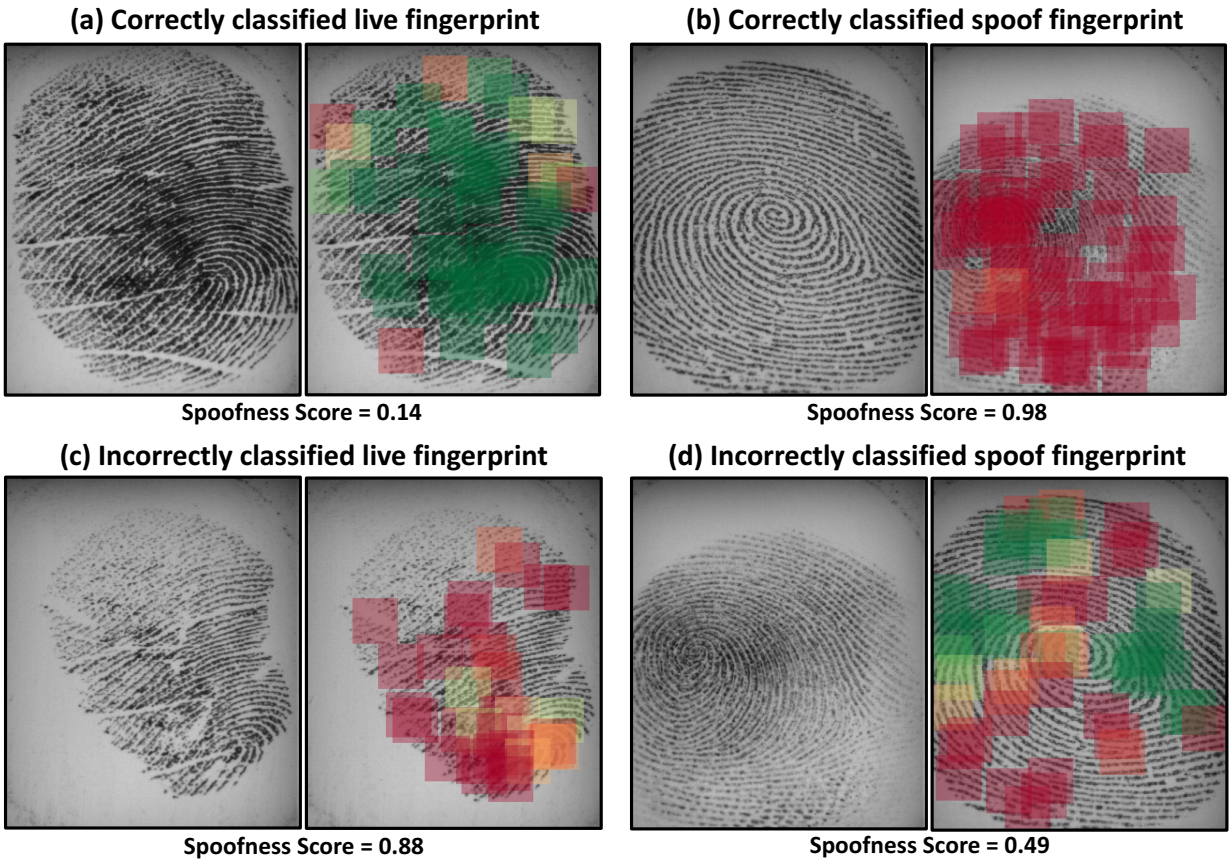


Figure 2.20 Example live and spoof fingerprints for Biometrika sensor from LivDet 2015 dataset, correctly and incorrectly classified by our proposed approach. (Best viewed in color)

sidered, and (iv) score-level fusion of multi-resolution local patches boosts the spoof detection performance.

2.6.3.2 Intra-Sensor, Cross-Material

In this setting, the same sensor is used to capture all training and testing images, but the spoof images in the testing set are fabricated using new materials that were not seen during training. For the first set of cross-material experiments, we utilize (i) the LivDet 2017 dataset which contains three completely different spoof materials in the testing for each sensor, *i.e.*, Gelatine, Latex, and Liquid Ecoflex, and (ii) the LivDet 2015 dataset which contains two new spoof materials in the testing set for each sensor, *i.e.*, Liquid Ecoflex and RTV for Green Bit, Biometrika, and Digital Persona sensors, and OOMOO and Gelatin for Crossmatch sensor. The performance of the proposed ap-

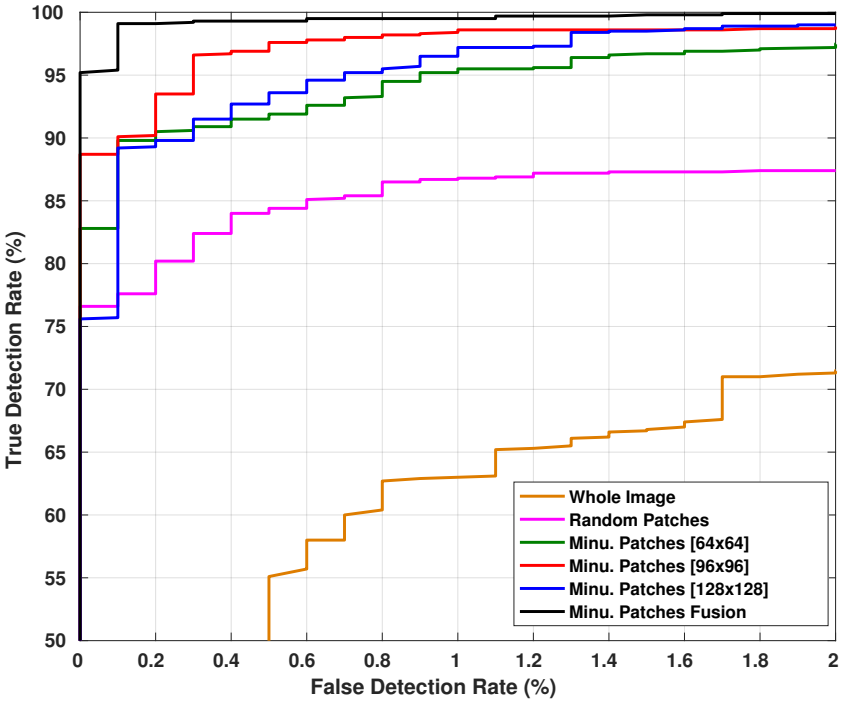


Figure 2.21 ROC curves for live v. spoof classification of fingerprint images from LivDet 2011 Dataset (Biometrika sensor) utilizing (i) whole image, (ii) randomly selected patches [96×96], (iii) minutiae-based patches of size [$p \times p$], $p \in \{64, 96, 128\}$, (iv) score-level fusion of multi-resolution patches. (Best viewed in color)

proach on cross-material experiments for LivDet 2017 and LivDet 2015 datasets are presented in Table 2.11 and Table 2.8 (column $Ferr fake\_unknown$), respectively, and is compared with the state-of-the-art performance reported in [113, 114]. A significant reduction in the error rate is achieved by the proposed method. For better generalizability, a second set of cross-material experiments are performed on LivDet 2011 and LivDet 2013 datasets, following the protocol adopted by the winner of LivDet 2015 [119]. Table 2.12 presents the achieved error rates on these experiments, along with the spoof fabrication materials used in training and testing sets.

2.6.3.3 Cross-Sensor Evaluation

In this evaluation, the training and the testing images are obtained from two different sensors but from the same dataset. This setting reflects the algorithm's strength in learning the common characteristics used to distinguish live and spoof fingerprints across fingerprint acquisition devices. For

Table 2.11 Performance comparison between the proposed approach and state-of-the-art results [114] reported on LivDet 2017 dataset for cross-material experiments in terms of Average Classification Error (ACE) and Ferrfake @ Ferrlive = 1%.

LivDet 2017 Dataset [114]	**State-of-the-Art**	**Proposed Approach**	
	ACE (%)	**ACE(%)**	**Ferrfake @ Ferrlive = 1%**
GreenBit	2.94	**2.33**	6.57
Orcanthus	**5.84**	7.04	26.05
Digital Persona	4.41	**2.90**	20.32
Average (LivDet 2017 Winner)	4.40 (4.75)	**4.09**	17.65

Table 2.12 Performance comparison between the proposed approach and state-of-the-art results reported on LivDet 2011 and LivDet 2013 datasets for cross-material experiments, in terms of Average Classification Error (ACE) and Ferrfake @ Ferrlive = 1%.

Dataset	**Spoof Materials**		**State-of-the-Art**	**Proposed Approach**	
	Materials - Training	**Materials - Testing**	**ACE (%)**	**ACE (%)**	**Ferrfake @ Ferrlive = 1%**
Biometrika 2011	EcoFlex, Gelatine, Latex	Silgum, WoodGlue	10.10 [119]	**4.60**	8.15
Biometrika 2013	Modasil, WoodGlue	EcoFlex, Gelatine, Latex	2.10 [126]	**1.30**	0.34
ItalData 2011	EcoFlex, Gelatine, Latex	Silgum, WoodGlue, Other	7.00 [126]	**5.20**	7.80
ItalData 2013	Modasil, WoodGlue	EcoFlex, Gelatine, Latex	1.25 [126]	**0.60**	0.68
Average			5.11	**2.93**	4.24

instance, using the LivDet 2011 dataset, images from the Biometrika sensor are used for training, and the images from ItalData sensor are used for testing. We follow the protocol for selection of training and testing sets for cross-sensor and cross-dataset experiments as adopted by Nogueira et al. [119]. Table 3.7 compares the average classification error and Ferrfake @ Ferrlive = 1% for the proposed approach with the state-of-the-art results obtained by [119] and [126] on cross-sensor experiments.

2.6.3.4 Cross-Dataset Evaluation

In this scenario, the training and the testing images are obtained using the same sensor, but from two different datasets, (*i.e.*, only the capture environments are different). For instance, training images are acquired using the Biometrika sensor from LivDet 2011 dataset and the testing images are acquired using the Biometrika sensor from LivDet 2013. This set of experiments captures the algorithm's invariance to the changes in environment for data collection. Table 2.14 presents the average classification error and Ferrfake @ Ferrlive = 1%. Results in Table 2.14 show that the

Table 2.13 Performance comparison between the proposed approach and state-of-the-art results [119] reported on LivDet 2011 and LivDet 2013 datasets for cross-sensor experiments, in terms of Average Classification Error (ACE), and Ferrfake @ Ferrlive = 1%.

Training Dataset (Testing Dataset)	**State-of-the-Art**	**Proposed Approach**	
	ACE (%)	**ACE (%)**	**Ferrfake (%) @ Ferrlive = 1%**
Biometrika 2011 (ItalData 2011)	29.35 [126]	**25.35**	50.81
ItalData 2011 (Biometrika 2011)	27.65 [126]	**25.21**	76.20
Biometrika 2013 (ItalData 2013)	**1.50** [126]	4.30	12.73
ItalData 2013 (Biometrika 2013)	**2.30** [119]	3.50	70.35
Average	15.20	**14.59**	**52.52**

Table 2.14 Performance comparison between the proposed approach and state-of-the-art results [126] reported on LivDet 2011 and LivDet 2013 datasets for cross-dataset experiments, in terms of Average Classification Error (ACE) and Ferrfake @ Ferrlive = 1%.

Training Dataset (Testing Dataset)	**State-of-the-Art**	**Proposed Approach**	
	ACE (%)	**ACE (%)**	**Ferrfake (%) @ Ferrlive = 1%**
Biometrika 2011 (Biometrika 2013)	14.00 [126]	**7.60**	89.60
Biometrika 2013 (Biometrika 2011)	34.05 [126]	**31.16**	78.84
ItalData 2011 (ItalData 2013)	8.30 [126]	**6.70**	16.70
ItalData 2013 (ItalData 2011)	44.65 [126]	**26.16**	75.09
Average	25.25	**17.91**	65.06

proposed local patch based approach achieves a reduction of 29% in the average classification error from 25.25% in [126] to 17.91% in our approach. However, the average Ferrfake @ Ferrlive = 1% that we report is 52.52% and 65.06% for cross-sensor and cross-dataset scenarios respectively, indicating the challenges, especially in applications where a high level of spoof detection accuracy is needed.

2.6.3.5 Government Controlled Tests

The evaluation scenario of GCT-I, GCT-II, and GCT-III is similar to a cross-dataset evaluation, as we utilize the same fingerprint reader for collecting training and testing data, but in different environments. The training data is collected in a lab environment at MSU whereas testing dataset are collected in a simulated operational setting at JHU facilities in Maryland. Table 2.15 presents the achieved PA True Detection Rate (%) @ False Detection Rate = 0.2%. The selection of this

Table 2.15 True Detection Rate (%) @ False Detection Rate = 0.2% on the GCT-I, GCT-II, and GCT-III evaluation datasets.

Dataset	**Proposed Approach**
GCT-I	**TDR (%) @ FDR = 0.2%**
CrossMatch Guardian 200	99.60
Lumidigm Venus 302	97.44
GCT-II	
CrossMatch Guardian 200	99.20
GCT-III	
CrossMatch Guardian 200	99.81

metric is based on the requirements of IARPA ODIN program [123] and represents the percentage of PAs able to breach the biometric system security when the reject rate of legitimate users $\leq 0.2\%$.

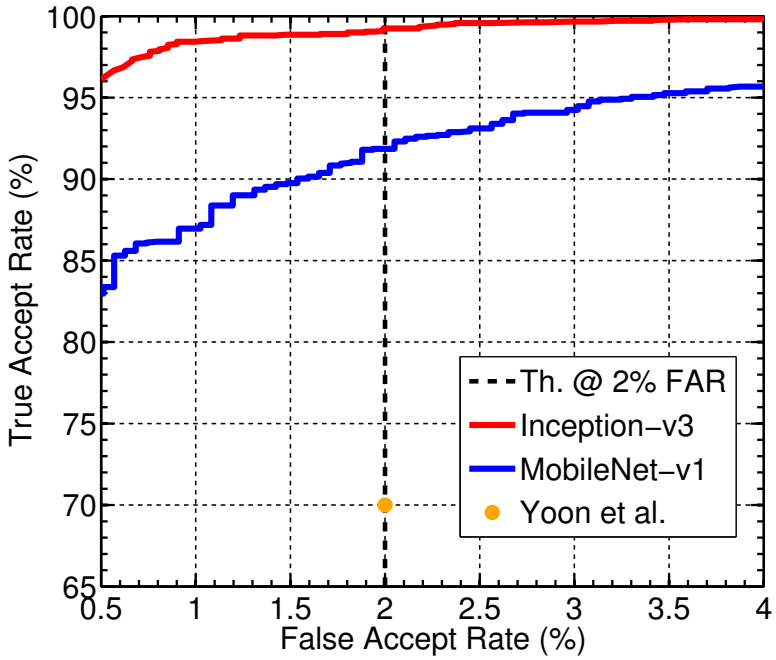


Figure 2.22 Performance curves for the proposed altered fingerprint detection approach utilizing Inception-v3 and MobileNet-v1 CNN models. Yoon et al. [170] (baseline) achieved a TDR of 70% @ FDR = 2% on 4,433 altered fingerprints, while the proposed approach achieves a TDR (over five folds) of 99.24% $\pm$ 0.58% @ FDR = 2% on 4,815 altered fingerprints. (Best viewed in color)

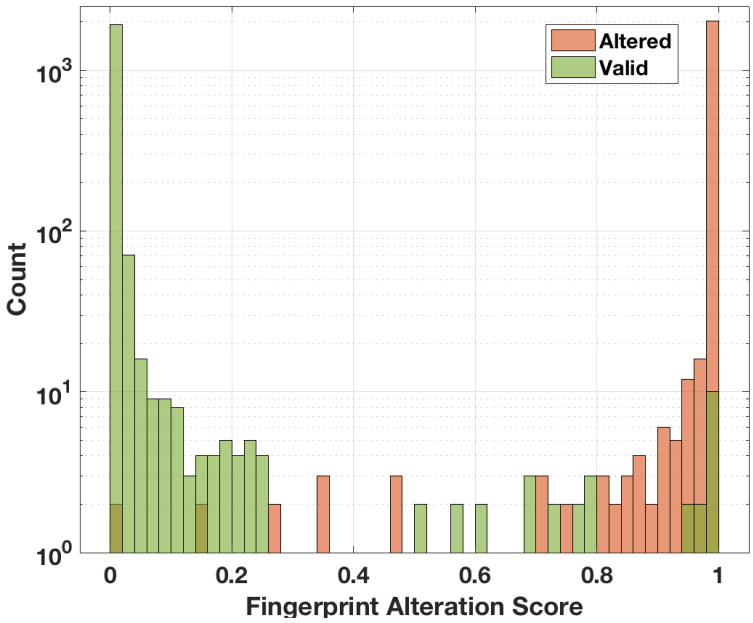


Figure 2.23 Alteration score histograms for bonafide and altered fingerprints obtained by the proposed approach using the best performing Inception-v3 model. The small overlap between the bonafide and altered score distributions is an indication of high discrimination power of the model. Note that the Y-axis is presented in log scale. (Best viewed in color)

2.6.4 Altered Fingerprint Detection and Localization

Figure 2.22 shows the Receiver Operating Characteristic (ROC) curves for the proposed altered fingerprint detection approach (Inception-v3 and MobileNet-v1) compared with state-of-the-art [170]. The red curve shows the accuracy of the Inception-v3 implementation and the blue curve shows the accuracy of the MobileNet-v1 implementation. Inception-v3 outperforms MobileNet-v1 architecture ($\sim 99\%$ to $\sim 92\%$), while the computational requirement[18] for MobileNet-v1 (6 ms) is almost 10 times lower compared to time required by the Inception-v3 architecture (50 ms). The superior performance of Inception-v3 over Mobilenet-v1 network can be attributed to (i) the deeper convolutional network providing higher discrimination power and (ii) the larger input image size, 299×299 for Inception-v3 compared to 224×224 for Mobilenet-v1. Both network models show better detection performance than Yoon and Jain [170] which had a true detection rate of only 70.2% at a false positive rate of 2%.

[18]We utilized NVIDIA GTX 1080 Ti GPU to run our implementation of Inception-v3 and MobileNet-V1 based altered fingerprint detection.

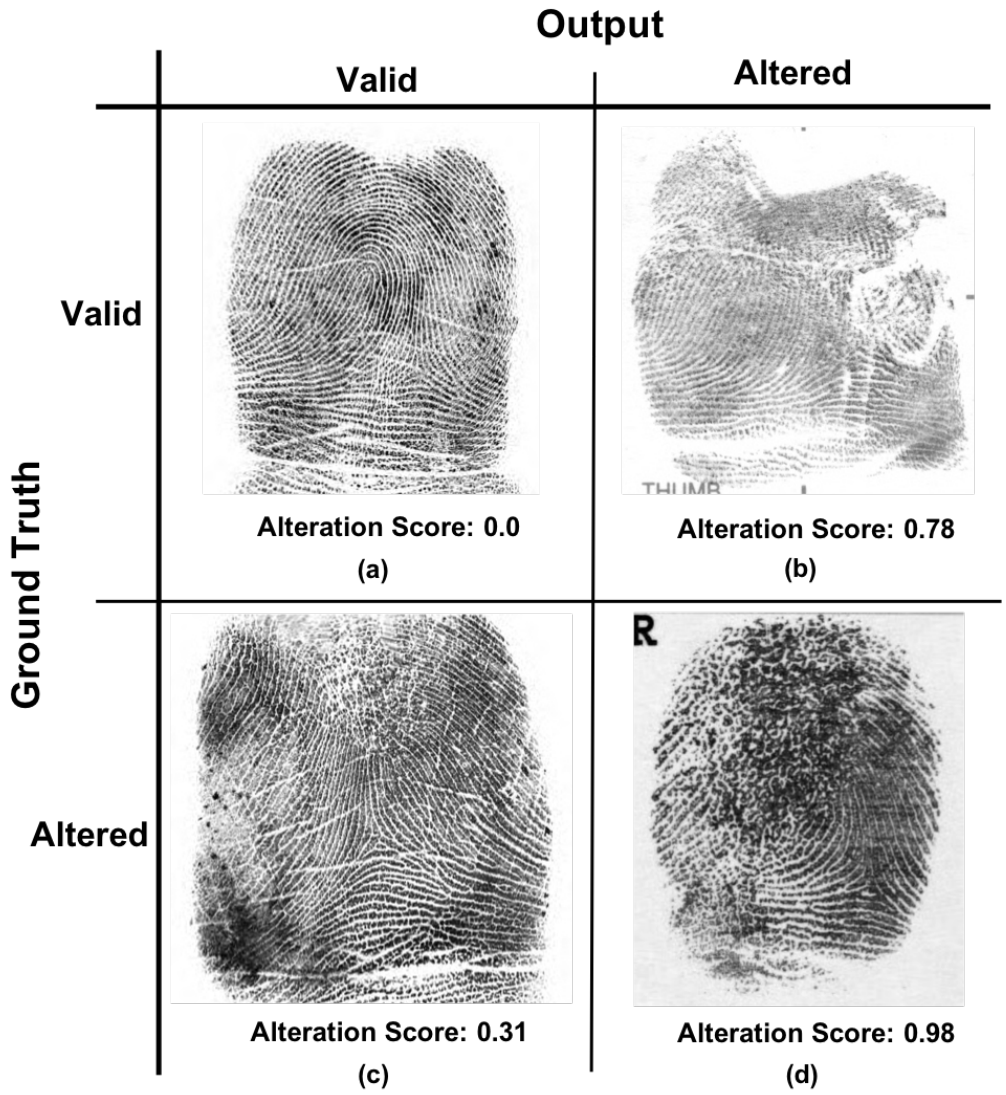


Figure 2.24 Example detections and their alteration scores output by the proposed approach. (a) and (d) present correctly classified images, while (b) and (c) present incorrect classifications. (b) a bonafide fingerprint that receives a high alteration score primarily due to the noisy region on the right. (c) contains a small region of alteration which is similar to the noise present in bonafide fingerprints.

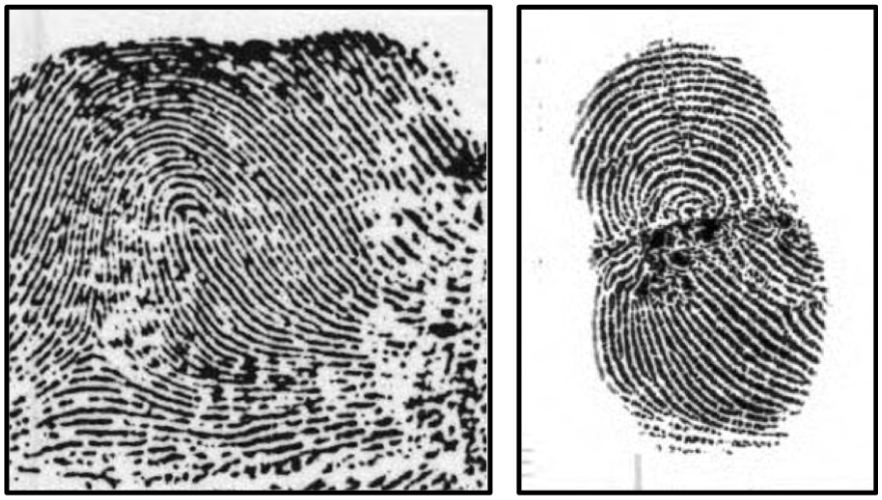


Figure 2.25 Example images with possible ground truth labeling error. (a) Incorrectly labeled as altered, and (b) incorrectly labeled as bonafide. The Inception-v3 model outputs an alteration score of 0.20 and 0.97 for (a) and (b), respectively, indicating (a) as bonafide and (b) as altered.

Figure 2.23 shows the histograms of scores produced by our Inception-v3 model for bonafide and altered fingerprint images. The very small overlap of the two distributions is an indication of the high accuracy of our model. We further investigated the images that were incorrectly labeled by our model according to the ground truth labels given at the time of training. Our visual inspection of these images suggests that some of images labeled as bonafide, look like altered fingerprints. This could be due to intentional alteration or cases of poor quality where fingerprint characteristics are degraded because of age or occupation (bricklayers, for example, are known to have poor quality fingerprints because their skin is severely damaged). On the other hand, some of the images labeled as altered, have a relatively small portion of the image as altered and most parts of the image look bonafide. In other words, most of the failure cases are due to the subjectivity of the labeling process. Example images of correct and incorrect classifications by the Inception-v3 model are shown in Figure 2.24 along with the scores generated by our model. Examples of incorrect ground truth labels are shown in Figure 2.25.

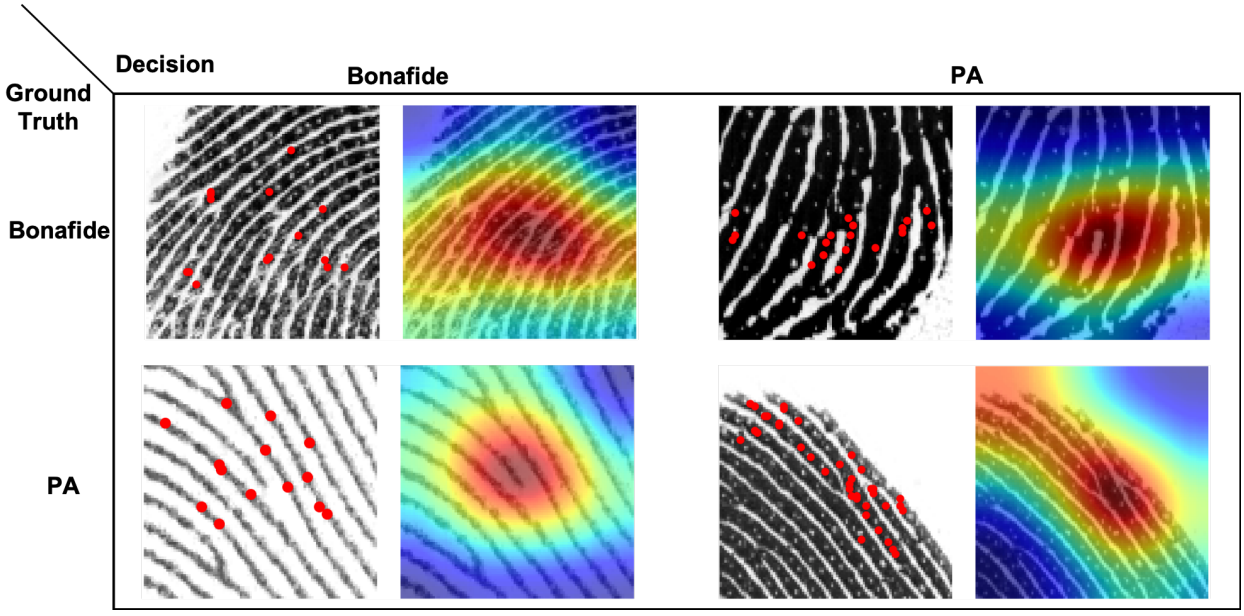


Figure 2.26 A confusion matrix of correct and incorrect classifications of bonafide and PA patches. The crucial regions that are responsible for the prediction made by the CNN architecture (CNN-Fixations) and the corresponding density heatmaps are illustrated on each local patch.

To evaluate the localization of fingerprint alterations, a two-fold cross validation is performed. Two Inception-v3 networks are trained using $81,969$ bonafide and $89,979$ altered patches, achieving an average EER of 8.5%.

2.7 Visualizing CNN Learnings

The use of convolutional neural networks (CNNs) has revolutionized computer vision and machine learning research achieving unprecedented performance in many tasks. But such solutions are usually considered as "black boxes" shedding little light on how they achieve high performance. One way to gain insights into what CNNs learn is through visual exploration, *i.e.*, to identify the image regions that are responsible for the final predictions. Towards this goal, visualization techniques [112, 144, 146] have been proposed to supplement the class labels predicted by CNN, in our case bonafide or PA, with the discriminated image regions (or saliency maps) exhibiting class-specific patterns learned by CNN architectures. The visualization technique proposed in [112] exploits the learned feature dependencies between consecutive layers of a CNN to identify the dis-

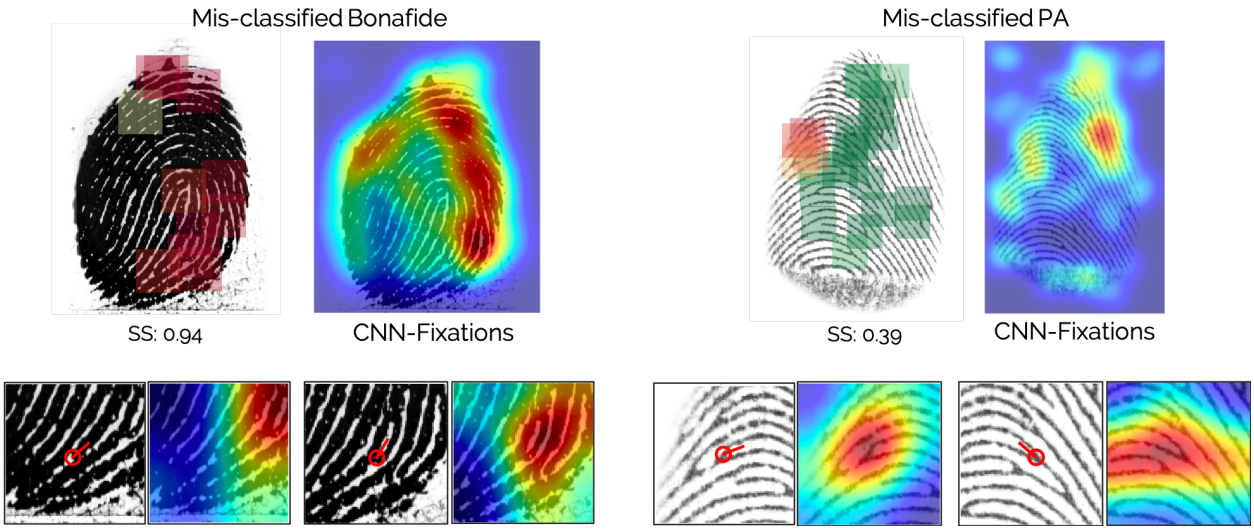


Figure 2.27 Examples of misclassified bonafide and PA fingerprint images along with the spoofness score (SS) output by the CNN architecture. Density heatmaps of the CNN-fixations are also presented.

criminative pixels, called *CNN-Fixations*, in the input image that are responsible for the predicted label. We utilize this visualization technique to understand the representation learning of our CNN models and identify the crucial regions in fingerprint patches responsible for final predictions. Figure 2.26 presents a confusion matrix of correct and incorrect classifications of bonafide and PA patches illustrating CNN-Fixations and the corresponding density heatmaps. We observe that there is a high density of fixations along friction ridge lines and at pore locations, suggesting that these are definitely crucial regions in distinguishing bonafide vs PA patches. Figure 2.27 presents additional examples of misclassified bonafide and PA fingerprint images along with a couple of local patches. In the case of bonafide whole image misclassified as PA, we observe a high density of points on the right edge of the image where the friction ridge lines are collapsed due to high moisture resulting in narrow valleys. In the case of misclassified PA image, the CNN-fixations exhibit a multi-modal distribution where the right region is dominating resulting in the average spoofness score of 0.39.

A deep convolutional neural network (CNN) is shown to be universal, implying that it can be used to approximate any continuous function to an arbitrary accuracy given the depth of the neural network is large enough [173]. “Instead of using a general filter bank, a neural network is trained

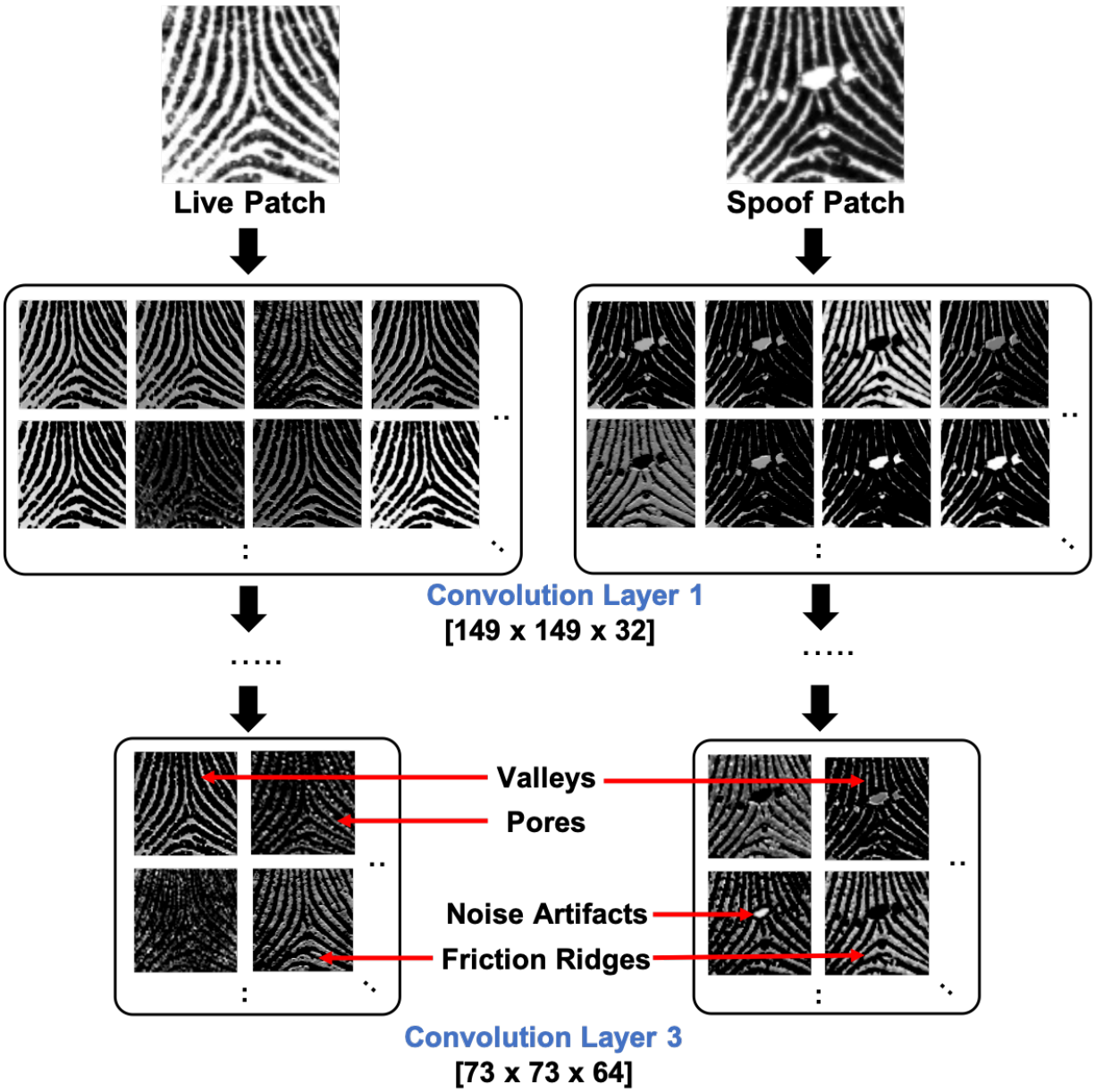


Figure 2.28 Illustration of the filter outputs, for a live and a spoof fingerprint patch, after the first and third convolution layers in the CNN architecture (Inception-v3). Different filters focus on different features such as location of sweat pores, noise artifacts, friction ridge, valley noise, etc.

to *find* a minimal set of specific filters, so that both the feature extraction and classification tasks are performed by the same unified network" [81]. After training the Fingerprint Spoof Buster, we use the same live and spoof fingerprint patches used in Figure 2.8 to visualize the filter outputs after the first and third convolution layers in the CNN architecture, shown in Figure 2.28. We observe that different filters focus on different features such as location of sweat pores, noise artifacts, friction ridge, valley noise, etc., marked in red. The CNN-architecture learns the non-linear complex relationship between the different features extracted at various scales from the input fingerprint image to achieve the high performance. It is however still an on-going research problem to understand and visualize the features learned by the CNN architectures.

2.8 Computing Times

The MobileNet-v1 CNN model takes around 6-8 hours to converge using a single Nvidia GTX 1080 Ti GPU with approximately $96,000$ local patches from $2,000$ fingerprint images ($2,000$ images $\times$ 48 patches/fingerprint image) in the training set. The average spoof detection time for an input image, including minutiae detection, local patch extraction and alignment, inference of spoofness scores for local patches, and producing the final spoof detection decision, is 100 ms using a Nvidia 1080Ti GPU and $1,500$ ms on a commodity smartphone.

2.9 Fingerprint Spoof Buster Lite

Fingerprint Spoof Buster evaluates all local patches corresponding to the detected minutiae. The individual scores output by the CNN model for each of the local patches is averaged to produce a global spoofness score. The time required to evaluate a single patch utilizing MobileNet-v1 CNN model on a commodity smartphone, such as Samsung Galaxy S8[19] (Qualcomm Snapdragon 835 64-bit Octa Core 2.35GHz Processor and 4GB RAM), is around 48ms. This results in an average execution time of 1.5 seconds per image (with an average no. of 35 minutiae/image). Moreover, a

[19] https://www.gsmarena.com/samsung_galaxy_s8-8161.php

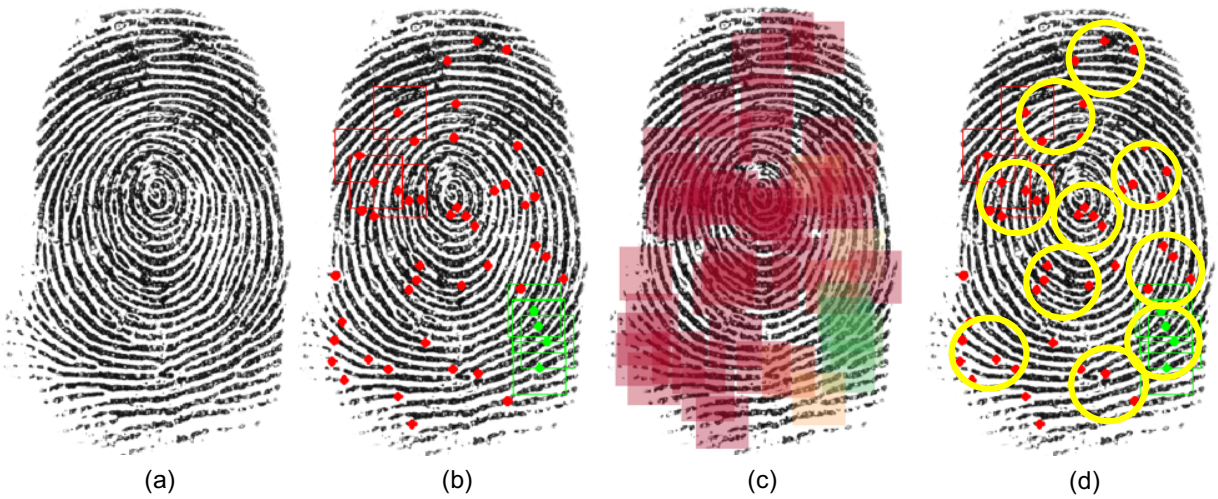


Figure 2.29 Minutiae clustering. (a) fingerprint image; (b) extracted minutiae overlaid on (a); (c) 96×96 patches centered at each minutiae; (d) minutiae clustering using k-means (k is set to 10 here). The clusters, highlighted as yellow circles of same size, are shown only for illustrative purposes. In practice, the cluster sizes may vary based on the minutiae distribution.

MobileNet-v1 trained model in ProtoBuf (.pb) format takes around 13MB. These computation and memory requirements are too large to yield an acceptable "real-time" spoof detection of a fraction of a second.

2.9.1 Proposed Optimizations

In order to reduce the memory and computation requirements for real-time operation on a commodity smartphone, we propose the following two optimizations:

Model Quantization: Tensorflow-lite[20] is used to convert the MobileNet-v1 (.pb) model to *tflite* format, resulting in a light-weight and low-latency model with weights quantized to perform byte computations instead of floating point arithmetic. The resultant model takes only 3.2MB of memory and can execute PAD for a single patch on s Samsung Galaxy S8 smartphone in around 10ms, approximately 80% reduction in computation and memory requirements.

Reduce the required number of inferences: It has been observed that minutiae points in a fingerprint image are distributed in a non-uniform manner [127]. This obviates the need for evaluating all minutiae-centered patches. We cluster the minutiae using K-means clustering [80] (see Figure 2.29

[20]https://www.tensorflow.org/lite/

Table 2.16 Detection time and PAD performance (TDR @ FDR = 0.2%) of Fingerprint Spoof Buster Lite.

# Minutiae Clusters	**Time Required (in ms) (Avg. $\pm$ s.d.)**	**TDR (%) @ FDR = 0.2%**
5	53 ± 10	93.9 ± 1.1
10	$\mathbf{98 \pm 8}$	$\mathbf{95.3 \pm 0.5}$
15	151 ± 11	95.3 ± 0.5
20	202 ± 10	95.3 ± 0.6
25	247 ± 24	95.7 ± 0.5
30	301 ± 25	95.7 ± 0.4
All Minutiae (avg. = 35)	$\mathbf{510 \pm 26}$	$\mathbf{95.7 \pm 0.1}$

Note: Samsung Galaxy S8 smartphone (Qualcomm Snapdragon 835 64-bit Octa Core 2.35GHz Processor and 4GB RAM) costs $\$349$.

(d)), extract a single patch (96×96) centered at the centroid of each cluster, and assign a weight to each cluster based on the number of members (minutiae points) that belong to that cluster. A cluster with large number of (minutiae) members is given a higher weight. The final spoofness score is computed as a weighted average of spoofness scores of centroid-based local patches. The weighted score-fusion is utilized to achieve a similar global spoofness score as obtained in the case when no clustering is performed.

Apart from the above two optimizations, we modify the MobileNet-v1 network such that the input image size is 96×96, the same size as the minutiae patch. Correspondingly, the kernel size used in the last average pool layer is reduced from 7×7 to 3×3. This reduces the time required to train the network on a dataset with around $100,000$ patches from 6-8 hours to 2-2.5 hours using a single NVIDIA GTX 1080Ti GPU, without any drop in PAD performance. We utilized the tensorflow-slim library[21] for our experiments.

Table 2.16 presents the accuracy of Fingerprint Spoof Buster Lite (TDR (%) @ FDR = 0.2%) and the average time required to evaluate minutiae-based patches on Galaxy S8. Since the output clusters from K-means clustering depend on the cluster initialization, we use 5-fold cross-validation and report average $\pm$ std. for both the evaluation time and PAD performance. Table 2.16 shows that a total of 10 minutiae clusters are suitable to maintain PAD performance (TDR = 95.3%

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

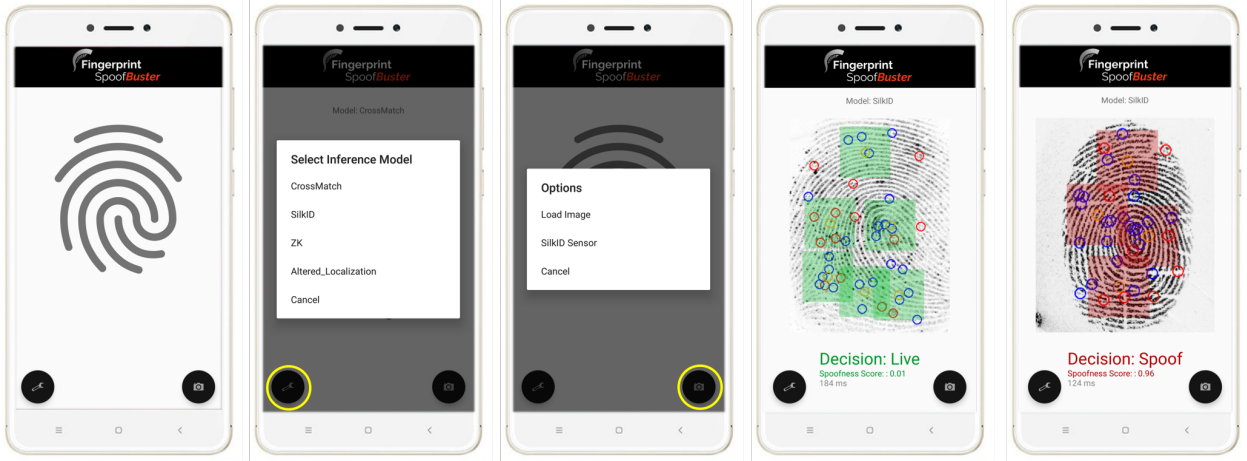


Figure 2.30 User interface of the Android application, *Fingerprint Spoof Buster Lite* shown in (a). It allows selection of an inference model as shown in (b). The user can load a fingerprint image from phone storage or capture a live scan from a fingerprint reader as shown in (c). The app executes PAD and displays the final decision along with highlighted local patches on the screen shown in (d) and (e).

compared to 95.7% @ FDR = 0.2%), while reducing the computational requirement by almost 80%.

2.9.2 Android Application

Given the reduction in resource requirements, an Android-based application (app) for Fingerprint Spoof Buster, called *Fingerpint Spoof Buster Lite*, was developed. The app provides an option to select an inference model trained on images from different fingerprint readers such as CrossMatch, SilkID[22], etc., as shown in Figure 2.30 (b). The app can evaluate a fingerprint image input by a fingerprint reader connected to the mobile phone via an OTG (on-the-go) cable. It also allows loading and evaluating an image from the phone storage/gallery (see Figure 2.30 (c)). The app displays the captured image with extracted fingerprint minutiae overlaid on the fingerprint image. Local patches centered around the centroid of minutiae clusters are evaluated and highlighted based on the spoofness score. After evaluation, the app presents the final decision (Live / Spoof), spoofness score, and PA detection time (see Figures 2.30 (d) and (e)).

[22]http://www.silkid.com/products/

2.10 Summary

A robust and accurate method for fingerprint presentation attack detection is critical to ensure the reliability and security of fingerprint authentication systems. In this chapter, we have utilized fingerprint domain knowledge by extracting local patches centered and aligned using minutiae in the input fingerprint image for fingerprint presentation attack detection. The local patch based approach, called Fingerprint Spoof Buster, provides salient cues to differentiate PA fingerprints from bonafide fingerprints. Spoof buster is able to achieve a significant reduction in the error rates for intra-sensor (63%), cross-material (43%), cross-sensor (4%), and cross-dataset scenarios (29%) compared to state-of-the-art on public domain datasets. A GUI is developed to allow an operator or system designer to analyze the input fingerprint images for bonafide and PA regions. We also trained a CNN model using operational datasets of $4,815$ altered and $4,815$ bonafide fingerprint images for altered fingerprint detection and localization. Our altered fingerprint detection model achieves a True Detection Rate (TDR) of 99.24% @ False Detection Rate (FDR) of 2%, compared to the previous state-of-the-art result of TDR = 70% at FDR = 2% which used a smaller operational dataset. Finally, we presented a light-weight version of the proposed PAD as an Android app that can run on a commodity smartphone (Samsung Galaxy S8) without significant drop in performance and make a PA detection in real-time (under 100ms).

Chapter 3

Fingerprint PAD Generalization

In the previous chapter, we tackled fingerprint presentation attack detection (PAD) by utilizing local patches (96×96) centered and aligned using fingerprint minutiae to train MobileNet-v1 and Inception-v3 models. This fusion of fingerprint domain knowledge (minutiae) and deep-learning based approaches provided state-of-the-art performance for fingerprint PAD. In this chapter, we address one of the major challenges of deep-learning based PAD approaches, namely, *fingerprint PAD generalization*. Our main focus is to improve *cross-material* and *cross-sensor* PAD generalization performance, while maintaining high performance in the known-material and known-sensor scenarios.

3.1 Introduction

New approaches to fingerprint PAD have proposed convolutional neural network (CNN) based solutions which have been shown to outperform hand-crafted features on publicly available LivDet databases [23, 24, 87, 110, 119, 126, 156]. However, one of the major limitations of existing PAD approaches is their poor generalization performance across "unknown" PA materials, that were not used during training. To generalize an algorithm's effectiveness across PA fabrication materials, called *cross-material* performance, PA detection has been referred to as an *open-set prob-*

Table 3.1 Summary of the studies primarily focused on fingerprint spoof generalization. The performance metrics utilized in different studies include ACE = Average Classification Error; EER = Equal Error Rate; and TDR = True Detection Rate (spoofs) @ a fixed FDR = False Detection Rate (spoofs).

Study	Approach	Database	Performance
Rattani et al. [135]	Weibull-calibrated SVM	LivDet 2011	EER = 19.70%
Ding & Ross [35]	Ensemble of multiple one-class SVMs	LivDet 2011	EER = 17.60%
Chugh & Jain [24]	MobileNet trained on minutiae-centered local patches	LivDet 2011-2015	ACE = 1.48% (LivDet 2015), 2.93% (LivDet 2011, 2013)
Chugh & Jain [26]	Identify a representative set of spoof materials to cover the deep feature space	MSU-FPAD v2.0, 12 spoof materials	TDR = 75.24% @ FDR = 0.2%
Engelsma & Jain [46]	Ensemble of generative adversarial networks (GANs)	Custom database with live and 12 spoof materials	TDR = 49.80% @ FDR = 0.2%
Gonzlez-Soler et al. [59]	Feature encoding of dense-SIFT features	LivDet 2011-2015	TDR = 7.03% @ FDR = 1% (LivDet 2015), ACE = 1.01% (LivDet 2011, 2013)
Tolosana et al. [156]	Fusion of two CNN architectures trained on SWIR images	Custom database with live and 8 spoof materials	EER = 1.35%
Gajawada et al. [50]	Style transfer from spoof to live images to improve generalization; requires few samples of target material	LivDet 2015, CrossMatch sensor	TDR = 78.04% @ FDR = 0.1%
Zhang et al. [172]	Slim-ResCNN + Center of Gravity patches	LivDet 2017	Avg. Accuracy = 95.25%
Proposed Approach	Style transfer between known spoof materials to improve generalizability against completely unknown materials	MSU-FPAD v2.0, 12 spoof materials & LivDet 2017	TDR = **91.78%** @ FDR = 0.2% (MSU-FPAD v2.0); Avg. Accuracy = **95.88%** (LivDet 2017)

lem[1] [135]. Table 3.1 presents a summary of the studies primarily focused on cross-material PAD generalization. Engelsma and Jain [46] proposed using an ensemble of generative adversarial networks (GANs) on live fingerprint images with the hypotheses that features learned by a discriminator to distinguish between real live and synthesized live fingerprints can be used to separate live fingerprints from PA fingerprints as well. One limitation of this approach is that the discriminator in the GAN architecture may learn many features related to structural noise added by the generative process. Such features are likely not present in the PAs fabricated with unknown materials.

[1]Open-set problems address the possibility of new classes during testing, that were not seen during training. Closed-set problems, on the other hand, evaluate only those classes that the system was trained on.

Although, it has been shown that some PA materials[2] are easier to detect (e.g. EcoFlex, Gelatin, Latex) than others (e.g. Wood Glue, Silgum) when left out from training [24], the underlying reasons are unknown. To understand and interpret the generalization performance against unknown PAs, we investigate material characteristics (two optical and two physical properties) correlated with cross-material performance and 3D t-SNE[3] feature embeddings [103] to identify a representative set of materials that should be included to train a robust PAD. We also propose two different approaches to improve the generalization performance. The main contributions of this chapter are:

1. Evaluated the generalization performance of Fingerprint Spoof Buster, a state-of-the-art CNN-based PAD approach, by employing leave-one-out approach on a large dataset of $5,743$ bonafide and $4,912$ PA images using 12 different PA materials.

2. Investigated the 3D t-SNE visualization and material characteristics (two physical and two optical) to identify a "representative set" of materials (Silicone, 2D paper, Play Doh, Gelatin, Latex Body Paint, and Monster Liquid Latex) that could almost cover the entire PA feature space.

3. Designed a style transfer-based wrapper, called Universal Material Generator (UMG), to improve the generalization performance of any fingerprint spoof detector against spoofs made from materials not seen during training. It attempts to synthesize impressions with style (texture) characteristics potentially similar to unknown spoof materials by interpolating the styles from known spoof materials.

4. Improved the cross-sensor spoof detection performance by synthesizing large-scale live and spoof datasets using only 100 live images from a new target sensor. Our approach for improving cross-material performance also improves the cross-sensor performance of two state-of-the-art spoof detectors.

[2]Fig. 2.1 illustrates the twelve different PA materials used in this study.

[3]The approach T-distributed Stochastic Neighbor Embedding (t-SNE) models each high-dimensional object by a two or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability [103].

5. Fabricated physical spoof artifacts using a mixture of known spoof materials to show that the synthetically generated images using fingerprint images of the same set of spoof materials correspond to an unknown material with similar style (texture) characteristics.

6. A dynamic PAD solution utilizing sequences of minutiae-based local patches to train a CNN-LSTM architecture with the goal of learning discriminative spatio-temporal features for fingerprint PA detection. The proposed approach improves the spoof detection performance from TDR of 81.65% to 86.20% @ FDR = 0.2% in cross-material scenario using a dataset of $26,650$ live captures from 685 subjects (1333 unique fingers) and $32,930$ PA frames from 7 PA materials (with 14 variants).

3.2 Databases used to investigate Fingerprint Generalization

The following datasets have been utilized in this study:

- **MSU Fingerprint Presentation Attack Database (FPAD) v2.0**

  A database of $5,743$ live and $4,912$ spoof images captured on CrossMatch Guardian 200[4], one of the most popular slap readers. The database is constructed by combining the publicly available MSU Fingerprint Presentation Attack Dataset v1.0 (MSU-FPAD v1.0) [24] and Precise Biometrics Spoof-Kit Dataset (PBSKD). Tables 3.2 and 3.4 presents the details of this database including the sensors used, 12 spoof materials, total number of fingerprint impressions, and the number of minutiae-based local patches for each material type. Fig. 2.1 presents sample fingerprint spoof images fabricated using the 12 materials.

- **LivDet 2017**

  LivDet 2017 [114] dataset is the most recent[5] publicly-available LivDet dataset, containing over $17,500$ fingerprint images. These images are acquired using three different fingerprint readers, namely, Green Bit, Orcanthus, and Digital Persona. Unlike other LivDet datasets,

[4]https://www.crossmatch.com/wp-content/uploads/2017/05/20160726-DS-En-Guardian-200.pdf
[5]The testing set of LivDet 2019 database has not yet been made public.

spoof fingerprint images included in the test set are fabricated using new materials (Wood Glue, Ecoflex, and Body Double), that are not used in the training set (Wood Glue, Ecoflex, and Body Double). Table 3.2 presents a summary of the LivDet 2017 dataset.

- **SilkID Fast Frame Rate Dataset** A large-scale fingerprint database of $26,650$ live frames from 685 subjects, and $32,930$ PA frames of 7 materials (14 variants) collected on SilkID SLK20R fingerprint reader is utilized in the evaluation of the proposed dynamic approach. This database is constructed by combining fingerprint images collected from two sources. First, as part of the IARPA ODIN program [123], a large-scale Government Controlled Test (GCT-3) was conducted at Johns Hopkins University Applied Physics Laboratory (JHUAPL) facility in Nov. 2019, where a total of 685 subjects with diverse demographics (in terms of age, profession, gender, and race) were recruited to present their real (live) as well as PA biometric data (fingerprint, face, and iris). The PA fingerprints were fabricated using 5 different materials (11 variants) and a variety of fabrication techniques, including use of dental and 3D printed molds. For a balanced live and PA data distribution, we utilize only right thumb and right index fingerprint images for the live data. Second, we collected PA data in a lab setting using dental molds casted with three different materials, namely, Ecoflex (with flesh tone pigment), Crayola Model Magic (red and white colors), and Dragon Skin (with flesh tone pigment). The details of the combined database are summarized in Table 3.3.

3.3 Understanding PAD Generalization

We adopt the leave-one-out protocol to simulate the scenario of encountering *unknown* materials with the goal of evaluating the generalization performance of Fingerprint Spoof Buster. One PA material out of the 12 types is left out from the training set which is then utilized during testing. This requires training a total of 12 different MobileNet-v1 models each time leaving out one of the 12 different PA types. The $5,743$ bonafide images are partitioned into training and testing such

Table 3.2 Summary of the MSU-FPAD-v2 and LivDet 2017 datasets. Spoof fingerprint images included in the test set of LivDet 2017 are fabricated using new materials that are not used in the training set.

Dataset	**MSU-FPAD v2 [26]**	**LivDet 2017 [114]**		
Fingerprint Reader	CrossMatch Guardian 200	GreenBit Dacty Scan 84C	Orcanthus Certis2 Image	Digital Persona U.are.U 5160
Image Size ($px.$) ($w \times h$)	800×750	500×500	$300 \times n^{\dagger}$	252×324
Resolution (dpi)	500	569	500	500
#Live (Train / Test)	$4,743$ / $1,000$	$1,000$ / $1,700$	$1,000$ / $1,700$	999 / $1,692$
#Spoof (Train / Test)	$4,912$ (leave-one-out)	$1,200$ / $2,040$	$1,180^{*}$ / $2,018$	$1,199$ / $2,028$
Known Spoof Materials (Training)	**Leave-one-out**: 2D Printed Paper, 3D Universal Targets, Conductive Ink on Paper, Dragon Skin, Gelatin, Gold Fingers, Latex Body Paint, Monster Liquid Latex, Play Doh, Silicone, Transparency, Wood Glue	Wood Glue, Ecoflex, Body Double		
Unknown Spoof Materials (Testing)		Gelatine, Latex, Liquid Ecoflex		

† Fingerprint images captured using Orcanthus reader have a variable height ($350 - 450px$) depending on the friction ridge content.
*A set of 20 Latex spoof fingerprints found in the training set of Orcanthus fingerprint reader were excluded in our experiments. Only Wood Glue, Ecoflex, and Body Double are expected to be in the training dataset.

that there are $1,000$ randomly selected bonafide images in the testing set and the remaining $4,743$ images are utilized in the training set.

3.3.1 Performance against Unknown Materials

Table 3.4 presents the performance of Fingerprint Spoof Buster against unknown presentation attacks in terms of TDR @ FDR = 0.2%. The weighted average generalization performance achieved by the PAD with the leave-one-out method is TDR = 75.24%, compared to TDR = 97.20% @ FDR = 0.2% when all PA material types are known during training. The PA materials Dragon Skin, Monster Liquid Latex, Transparency, 3D Universal Targets, and Conductive Ink on Paper are easily detected with a TDR $\geq 90\%$ @ FDR = 0.2% even when these materials are not seen by the models during training. On the other hand, PA materials such as PlayDoh, Gelatin, 2D Printed Paper, and Silicone are the most affected when not seen during training achieving a TDR $\leq 70\%$ @ FDR = 0.2%. To understand the reasons for this difference in performance for different materials, we study the material characteristics in the next section.

Table 3.3 Summary of the SilkID Fast Frame Rate fingerprint database collected at GCT-III as part of IARPA ODIN Program [123].

PA Material	Mold Type	# Presentations	# Frames
Ecoflex silicone			
Ecoflex 00-35, flesh tone pigment	Dental	757	7,570
Ecoflex 00-50, flesh tone pigment	3D Printed	138	1,380
Ecoflex 00-50, tan pigment	3D Printed	130	1,300
Gelatin			
Ballistic gelatin, flesh tone dye	3D Printed	50	500
Knox gelatin, clear	3D Printed	84	840
Third degree silicone			
Light flesh tone pigment	Dental	131	1,310
Tan pigment	Dental	98	980
Beige suede powder	Dental	43	430
Medium flesh tone pigment	Dental	36	360
Crayola Model Magic			
White color	Dental	910	9,100
Red color	Dental	308	3,080
Pigmented Dragon Skin (flesh tone)	Dental	452	4,520
Conductive Silicone	3D Printed	87	870
Unknown PA (JHU-APL)	3D Printed	67	670
Total PAs		**3,291**	**32,910**
Total Lives (685 subjects)		**2,665**	**26,650**

3.3.2 PA Material Characteristics

Table 3.4 shows that some of the PA materials are easier to detect than others, even when left out from training. To understand the reason for this, it is crucial to identify the relationship between different PA types in terms of their material characteristics. If we can group the PA materials based on shared characteristics, it can potentially be used to identify a set of representative materials to train a robust and generalizable model. For the given dataset of fingerprint images from 12 different spoof materials, we measured the following characteristics: (i) *optical properties*: Ultra Violet - Visible (UV-Vis) spectroscopy response and Fourier Transform Infrared (FT/IR) Spectroscopy response, and (ii) *mechanical properties*: material elasticity and moisture content. These material characteristics were selected based on our discussions with material science experts[6].

[6]Material resistivity would be an important characteristic when performing a similar analysis for capacitive fingerprint readers.

Table 3.4 Summary of the dataset and generalization performance (TDR (%) @ FDR = 0.2%) with leave-one-out method. A total of twelve models are trained where the material left-out from training is taken as the new material for evaluating the model.

Fingerprint Presentation Attack Material	**#Images**	**#Local Patches**	**Generalization Performance** (TDR (%) @ FDR = 0.2%)
Silicone	$1,160$	$38,145$	67.62
Monster Liquid Latex	882	$27,458$	94.77
Play Doh	715	$17,602$	58.42
2D Printed Paper	481	$7,381$	55.44
Wood Glue	397	$12,681$	86.38
Gold Fingers	295	$9,402$	88.22
Gelatin	294	$10,508$	54.95
Dragon Skin	285	$7,700$	97.48
Latex Body Paint	176	$6,366$	76.35
Transparency	137	$3,846$	95.83
Conductive Ink on Paper	50	$2,205$	90.00
3D Universal Targets	40	$1,085$	95.00
Total PAs	**4,912**	**144,379**	**Weighted* Average:** 75.24
Total Bonafide	**5,743**	**228,143**	

* The performance is weighted by the number of images for each material (similar to as performed for publicly- available LivDet Datasets).

3.3.2.1 Optical Properties

Ultra Violet - Visible (UV-Vis) spectroscopy: The UV-Vis response represents the absorption of monochromatic radiations by the material at different wavelengths (ultraviolet (200-400 nm) to visible spectrum (400-750 nm)). A peak in the UV-Vis response indicates that the material has high absorbance of the light at that given wavelength [130]. A Perkin Elmar Lambda 900 UV/Vis/NIR spectrometer[7] was used to measure the light absorbance property of materials shown in Figure 3.1.

Fourier Transform Infrared (FT/IR) Spectroscopy: The FT/IR response of a given material is a signature of its molecular structure. The molecules absorb frequencies that are characteristic of their structure, called resonant frequencies, i.e., the frequency of the absorbed radiation matches with the vibrational frequency [148]. An FT/IR signature is a graph of infrared light absorbance (or transmittance) on the Y-axis vs. frequency on the X-axis (measured in reciprocal centimeters, i.e., cm^{-1} or wave numbers). Figure 3.2 presents the FT/IR response of 12 different PA materials measured by Jasco FT/IR-4600 spectrometer[8]. The FT/IR spectrometer provided material response

[7]http://www.perkinelmer.com/category/uv-vis-spectroscopy-uv

[8]https://jascoinc.com/products/spectroscopy/ftir-spectrometers/models/ftir-4000-series/

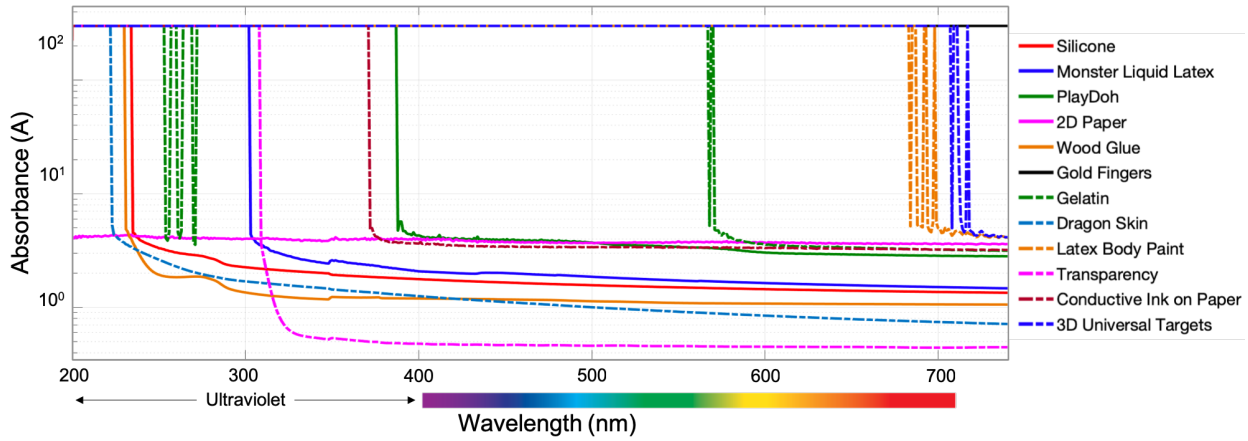


Figure 3.1 Light absorbance property of twelve PA materials in 200nm - 800nm wavelength spectrum computed using a Perkin Elmar Lambda 900 UV/Vis/NIR spectrometer [130].

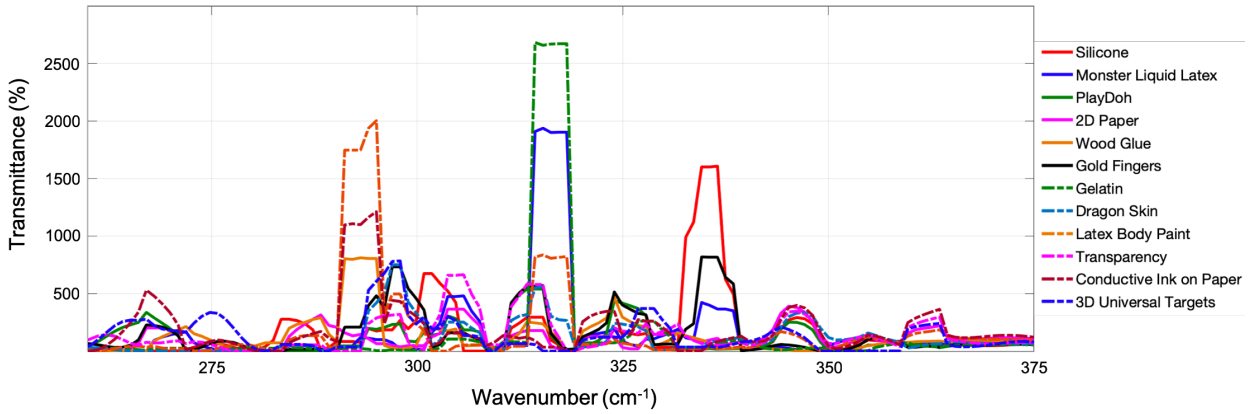


Figure 3.2 Fourier Transform Infrared Spectroscopy [148] of twelve PA materials in the 260 - 375 wavenumber range.

in the range $250 - 6,000$ wave numbers, but all the materials exhibited non-zero transmittance only in the range $250 - 375$ wave numbers.

3.3.2.2 Mechanical Properties

Material Elasticity: A fingerprint spoof fabricated using an elastic material undergoes higher deformation, resulting in large friction ridge distortion when the spoof is pressed against the fingerprint reader's glass platen, compared to less elastic materials. We classify the 12 different PA materials into three classes based on their observed elasticity: (i) *High elasticity*: Silicone, Monster Liquid Latex, Dragon Skin, Wood Glue, Gelatin, (ii) *Medium elasticity*: Play Doh, Latex Body Paint, 3D

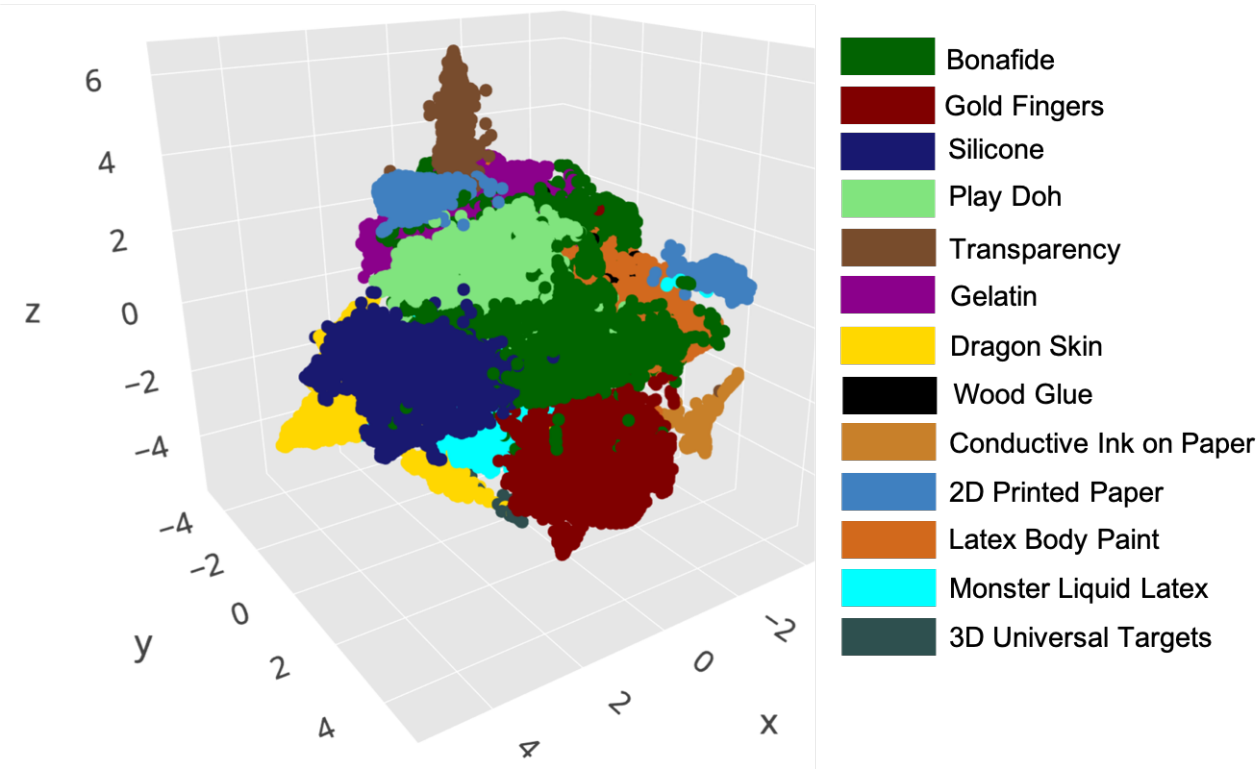


Figure 3.3 Representation of bonafide fingerprints and presentation attack instruments fabricated with different materials in the 3D t-SNE feature space. The original representation is 1024-dimensional obtained form the trained CNN model. (Best viewed in color). Available in 3D at https://plot.ly/~icbsubmission/0/livepa-feature-space/.

Universal Targets, and (iii) *Low elasticity*: 2D Paper, Gold Fingers, Transparency, and Conductive Ink on Paper.

Moisture Content: Another crucial material property is the amount of moisture content, which leads to varying degrees of contrast in the corresponding fingerprint image. PA materials with high moisture content (e.g. Silicone) produce high contrast images compared to materials with low moisture content (e.g. 2D Paper) on CrossMatch reader. We classify the 12 different PA materials into three classes of moisture content level based on the observed image contrast: (i) *High Moisture Level*: Silicone, Play Doh, Dragon Skin, (ii) *Medium Moisture Level*: Monster Liquid latex, Wood Glue, Gold Fingers, Gelatin, 3D Universal Targets, and (iii) *Low Moisture Level*: 2D Paper, Latex Body Paint, Transparency, Conductive Ink on Paper.

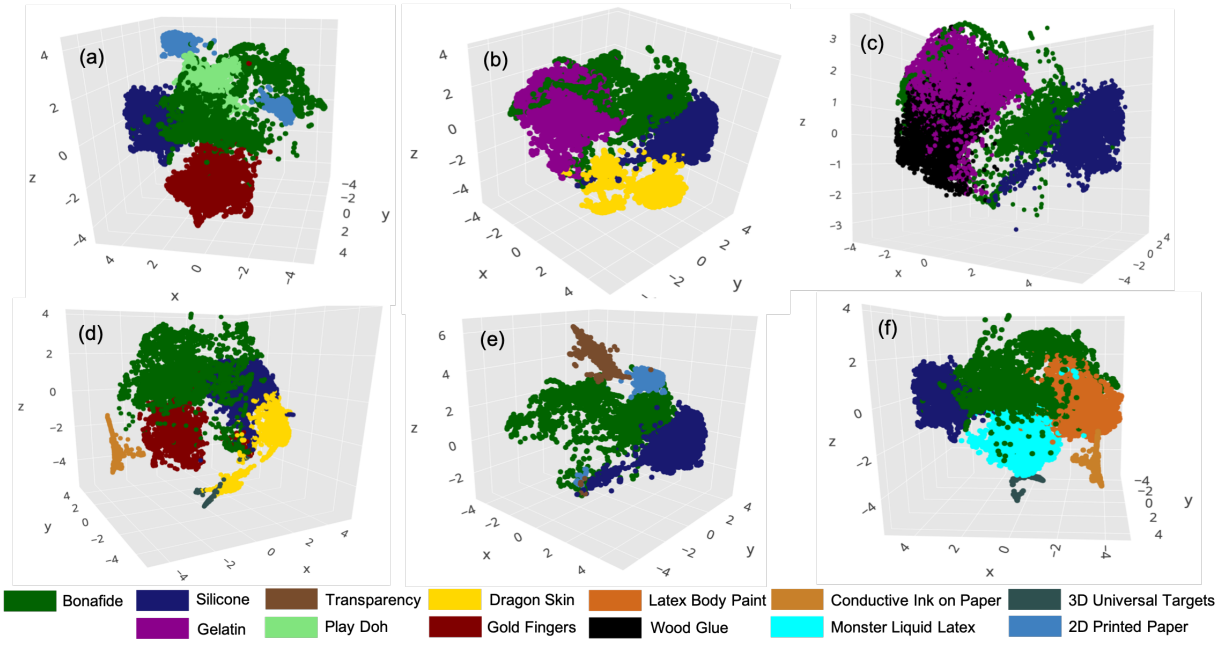


Figure 3.4 Representation of bonafide and different subsets of PA materials in 3D t-SNE feature space from different angles selected to provide the best view. The bonafide (dark green) and silicone (navy blue) are included in all graphs for perspective. (Best viewed in color)

3.3.3 3D t-SNE Visualization of Bonafide and PAs

To explore the relationship between bonafide and different PA materials, we train a single multi-class MobileNet-v1 model to distinguish between 13 classes, *i.e.*, bonafide and 12 PA materials. The training split includes a set of 100 randomly selected images or half the number of total images (whichever is lower) from each of the bonafide and PA materials for a total of $1,102$ images. In a similar manner, a test split is constructed from the remaining set of images for a total of $1,101$ images. This protocol is adopted to reduce the bias due to unbalanced nature of the training dataset. We extract the 1024-dimensional feature vector from the bottleneck layer of the MobileNet-v1 network [71] and project it to 3 dimensions using the t-SNE approach [103] (see Figure 3.3). Figures 3.4 (a)-(f) present the representation of bonafide and different subsets of PA materials in the 3D t-SNE feature space from different angles selected to provide a complete view. The Bonafide (dark green) and Silicone (navy blue) are included in all graphs for perspective. The 3D graph is generated using plotly library and is accessible at the link: https://plot.ly/~icbsubmission/0/livepa-feature-space/.

Figure 3.5 Average Pearson correlation values between 12 PA materials based on the material characteristics (two optical and two physical).

3.3.4 Representative Set of PA Materials

We utilize material characteristics and 3D t-SNE visualization to identify a set of representative materials to train a robust and generalizable model. From the four material characteristics, two continuous (*i.e.*, optical characteristics) and two categorical (*i.e.*, mechanical characteristics), we compute four 12×12 correlation matrices. For the two continuous variables, we compute the Pearson correlation[9] between all pairs of materials to generate two correlation matrices C^{uvvis} and C^{ftir}. For the two categorical variables, if two PA materials m_i and m_j belong to the same category, we assign $C_{i,j} = 1$, else $C_{i,j} = 0$, to generate $C^{elastic}$ and $C^{moisture}$. The four correlation matrices corresponding to each of the four individual material characteristics, are averaged to generate the final correlation matrix $C^{material}$, such that $C^{material}_{i,j} = (C^{uvvis}_{i,j} + C^{ftir}_{i,j} + C^{elastic}_{i,j} + C^{moisture}_{i,j})/4$, (see

[9]MATLAB's *corr* function is used to compute the Pearson correlation. https://www.mathworks.com/help/stats/corr.html

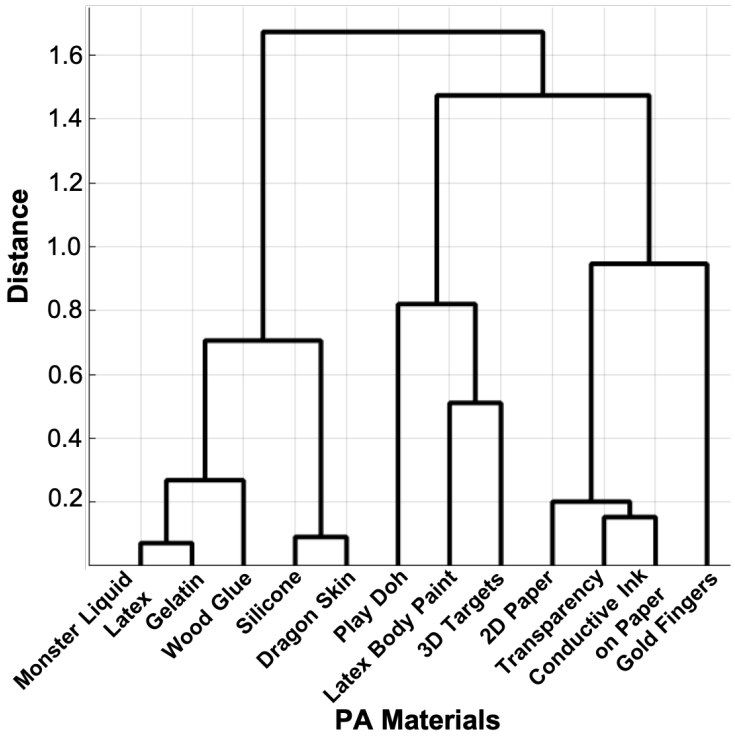


Figure 3.6 A complete-link dendrogram representing the hierarchical (agglomerative) clustering of PAs based on the shared material characteristics.

Figure 3.5) which is utilized to perform complete-link hierarchical (agglomerative) clustering[10] of the 12 PA materials. Figure 3.6 shows a complete-link dendrogram representing the hierarchical grouping of the 12 PA materials based on $C^{material}$. Based on the 3D t-SNE visualization and the hierarchical clustering of the 12 PA materials, we observe that:

- PA materials Silicone, Play Doh, Gelatin, and 2D Printed Paper are closest to Live fingerprints in the 3D t-SNE feature space compared to other materials. This explains why excluding any one of these materials in the training set resulted in poor generalization performance when tested against them. These PA materials appear in different clusters in the dendrogram (see Figures 3.4 (a) and 3.6).

- PA material Dragon Skin is easily detected when Silicone is included in training set as silicone is located between bonafide and Dragon Skin in the 3D t-SNE feature space (see Fig-

[10]We utilize MATLAB's *linkage* and *dendrogram* functions with parameters method='complete' and metric='correlation'.

ures 3.4 (b) and (d)). These materials, Dragon Skin and Silicone, also lie in the same cluster indicating shared material characteristics.

- PA material Transparency is easily distinguishable when 2D Printed Paper is included in training. In the t-SNE visualization, we observe that 2D Printed Paper appears in two different clusters, where one of the clusters is co-located with transparency (see Figures 3.4 (a) and (e)).

- PA materials Wood Glue and Gelatin are close to each other in 3D t-SNE feature space, potentially assisting each other if included in training (see Figure 3.4 (c)); whereas Gelatin is closer to Bonafide, which explains its worse performance compared to Wood Glue. These materials also form a second level cluster in the dendrogram.

- PA material Latex Body Paint is located between Bonafide and Conductive Ink on Paper, and PA material Monster Liquid Latex lies between Bonafide and 3D Universal Targets in 3D t-SNE visualization, which could explain the high detection for Conductive Ink on Paper and 3D Universal Targets (see Figure 3.4 (f)). However, these materials do not form a cluster until the last agglomeration step, indicating possibility of other material characteristics that could be further explored.

Based on these observations, we infer that a set of 6 PA materials (Silicone, 2D Paper, Play Doh, Gelatin, Latex Body Paint, and Monster Liquid Latex) almost covers the entire feature space around Bonafide (see Figure 3.4). A model trained using bonafide and these 6 PA materials achieved an average TDR = 89.76% $\pm$ 6.97% @ FDR = 0.2% when tested on each of the remaining 6 materials. This performance is comparable to the average TDR = $90.97\% \pm 7.27\%$ @ FDR = 0.2% when 11 PA materials and bonafides are used for training, indicating no significant contribution provided by including all the 11 PA materials in training. We posit that the PAD performance against a new material can be estimated by analyzing its material characteristics instead of collecting large datasets for each of the new materials.

3.4 Improving PAD Generalization

It has been shown that the selection of PA materials used in training (known PAs) directly impacts the performance against unknown PAs. In the previous section, we analyzed the material characteristics of 12 different spoof materials to identify a representative set of six materials that cover most of the PA feature space. Although, this approach can be used to identify if including a new PA material in training dataset would be beneficial, it does not improve the generalization performance against materials that are unknown during training. Moreover, with the increasing popularity of fingerprint authentication systems, hackers are constantly devising new fabrication techniques and novel materials to attack them. Also, it is prohibitively expensive to include all PA fabrication materials in training a PA detector.

Additionally, fingerprint images captured using different fingerprint sensors typically have unique characteristics due to different sensing technologies, sensor noise, and varying resolution. As a result, fingerprint PA detectors are known to suffer from poor generalization performance in the cross-sensor scenario, where the PAD is trained on images captured using one sensor and tested on images from another. Improving cross-sensor PA detection performance is important in order to alleviate the time and resources involved in collecting large-scale datasets with the introduction of new sensors. Next, we present two different approaches to improve the generalization performance of existing PAD solutions.

3.4.1 Universal Material Generator

In this section, we propose a style-transfer based wrapper, called *Universal Material Generator* (UMG), to improve the cross-material and cross-sensor generalization performance of fingerprint PA detectors against PAs made from materials not seen during training [28]. In particular, for the cross-material scenario, we hypothesize that the texture (style) information from the known PA types can be transferred from one type to another type to synthesize images potentially similar to novel PAs fabricated from materials, not seen in the training set. We posit that the synthesized PA

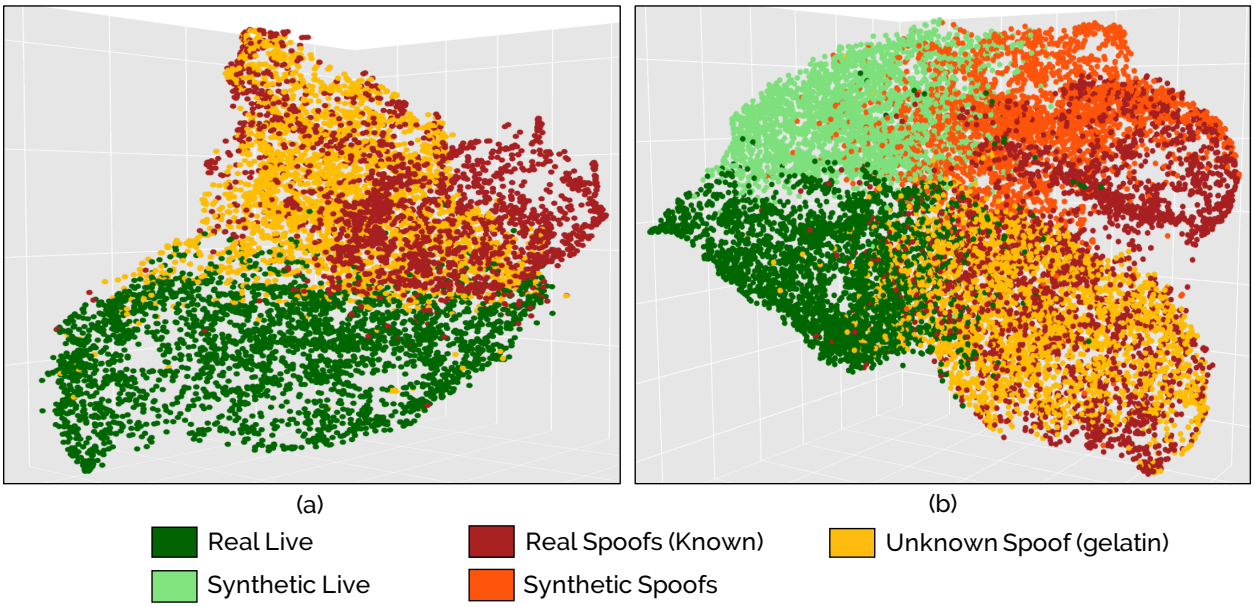


Figure 3.7 3D t-SNE visualization of feature embeddings learned by Fingerprint Spoof Buster [24] of (a) live (dark green) and eleven known PA materials (red) (2D printed paper, 3D universal targets, conductive ink on paper, dragon skin, gold fingers, latex body paint, monster liquid latex, play doh, silicone, transparency, and wood glue) used in training, and unknown PA, gelatin (yellow). A large overlap between unknown PA (gelatin) and live feature embeddings indicate poor generalization performance of state of the art PA detectors. (b) Synthetic live (bright green) and synthetic PA (orange) images generated by the proposed Universal Material Generator (UMG) wrapper improve the separation between real live and real PA. 3D t-SNE visualizations are available at http://tarangchugh.me/posts/umg/index.html (Best viewed in color)

images may occupy the space between the images from known PA materials in the deep feature space. Synthetic live fingerprint images are also added to the training dataset to force the CNN to learn generative-noise invariant features which discriminate between lives and PAs. In the cross-sensor scenario, we utilize a small set of only 100 bonafide fingerprint images from the target sensor, say Green Bit, and transfer its sensor-specific style characteristics to large-scale live and PA datasets available from a source sensor, say Digital Persona. Reusing large-scale PA datasets from existing sensors will reduce the steep cost associated with collecting large-scale bonafide and spoof databases for every new sensor.

The proposed UMG framework is used to augment CNN-based PA detectors, significantly improving their performance against novel materials, while retaining their performance on known materials. See Figure 3.10 for examples of some of the style transferred images.

3.4.1.1 Related Work

Realistic image synthesis is a challenging problem. Early non-parametric methods faced difficulty in generating images with textures that are not known during training [18]. Machine learning has been very effective in this regard, both in terms of realism and generality. Gatys et al. [53] perform artistic style transfer, combining the content of an image with the style of any other by minimizing the feature reconstruction loss and a style reconstruction loss which are based on features extracted from a pre-trained CNN. While this approach generates realistic looking images, it is computationally expensive since each step of the optimization requires a forward and backward pass through the pre-trained network. Other studies [88, 95, 160] have explored training a feed-forward network to approximate solutions to this optimization problem. There are other methods based on feature statistics to perform style transfer [73, 158]. Elgammal et al. [39] applied GANs to generate artistic images. Isola et al. [76] used conditional adversarial networks to learn the loss for image-to-image translation. Xian et al. [166] learned to synthesize objects consistent with texture suggestions. The proposed Universal Material Generator builds on [73] and is capable of producing realistic fingerprint images containing style (texture) information from images of two different PA materials. Existing style transfer methods condition the source image with target material style. However, in the context of fingerprint synthesis, this results in a loss in fingerprint ridge-valley information (*i.e.*, content). In order to preserve both style and content, we use adversarial supervision to ensure that the synthesized images appear similar to the real fingerprint images.

3.4.1.2 Proposed Approach

This approach includes three stages: (i) training the Universal Material Generator (UMG) wrapper using the PA images of known materials (with one material left-out from training), (ii) generating synthetic PA images using randomly selected image pairs of different but known materials, and (iii) training a PA detector on the augmented dataset to evaluate its performance on the "unknown" material left out from training. In all our experiments, we utilize local image patches (96×96) centered and aligned using minutiae location and orientation, respectively [24]. During the evaluation

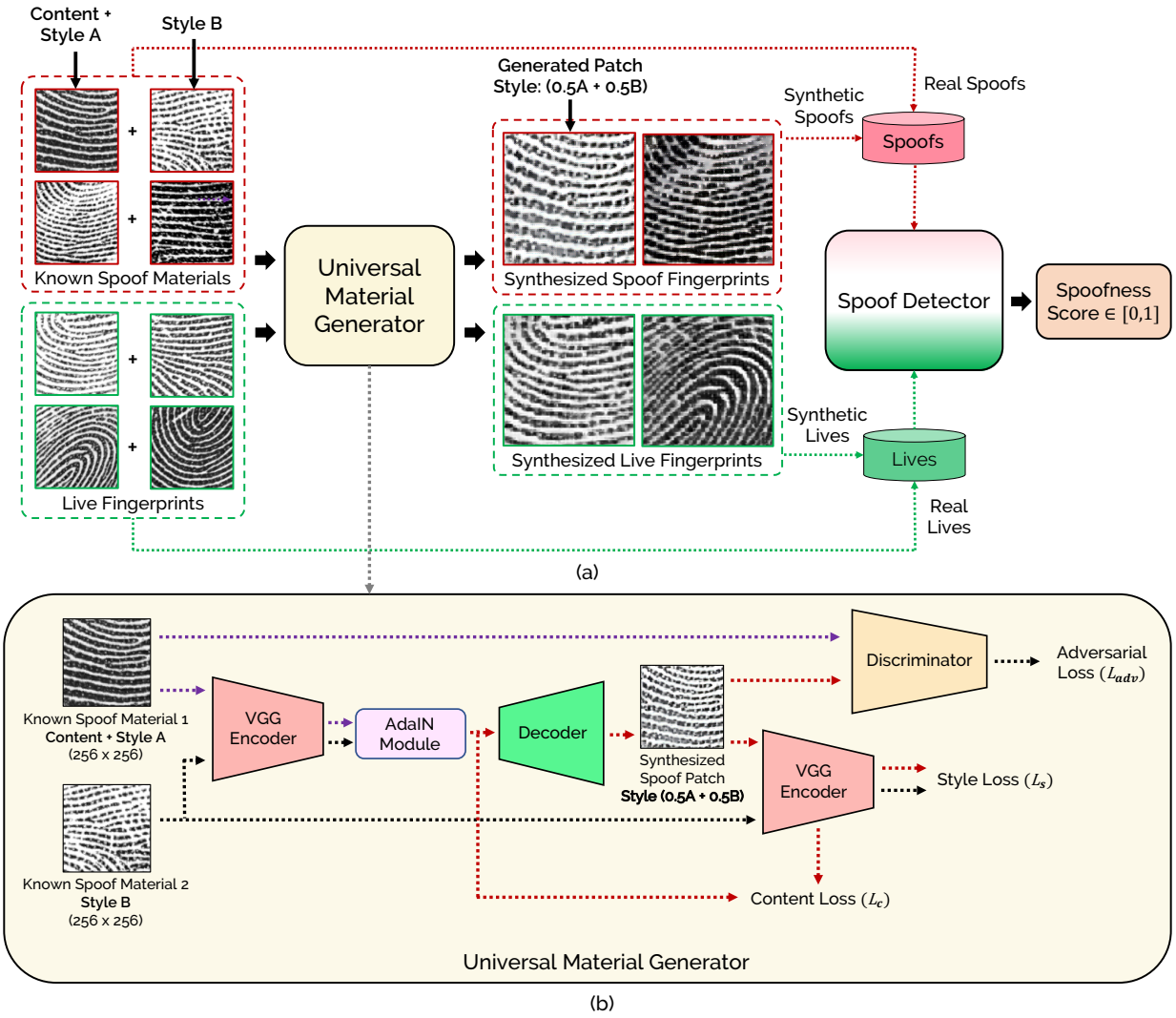


Figure 3.8 Proposed approach for (a) synthesizing PA and live fingerprint patches, and (b) design of the proposed Universal Material Generator (UMG) wrapper. An AdaIN module is used for performing the style transfer in the encoded feature space. The same VGG-19 [147] encoder is used for computing content loss and style loss. A discriminator similar to the one used in DC-GAN [133] is used for computing the adversarial loss. The synthesized patches can be used to train any fingerprint PA detector. Hence, our approach is referred to as a wrapper which can be used in conjunction with any PA detector.

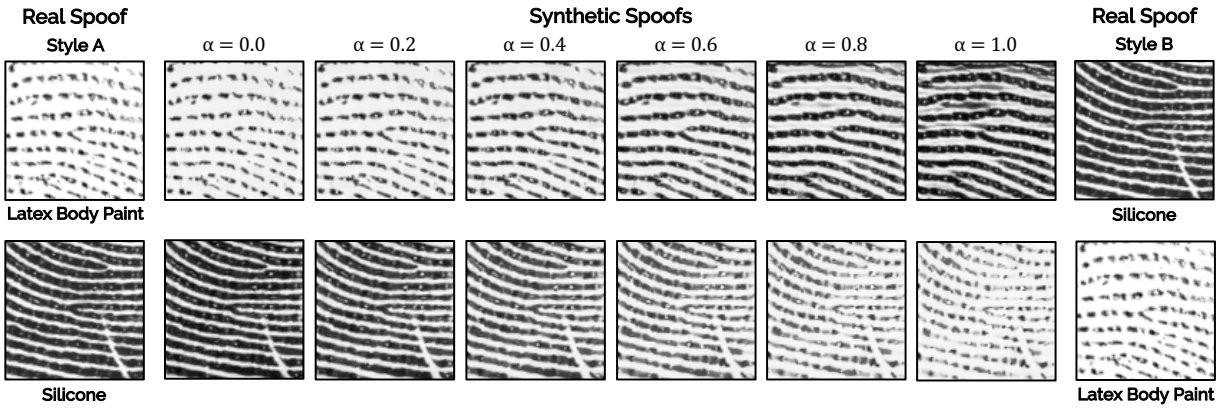


Figure 3.9 Style transfer between real PA patches fabricated with latex body paint and silicone to generate synthetic PA patches using the proposed Universal Material Generator (UMG) wrapper. The extent of style transfer can be controlled by the parameter $\alpha \in [0, 1]$.

stage, the PA detection decision is made based on the average of spoofness scores for individual patches output from the CNN model. An overview of the proposed approach is presented in Fig. 3.8.

The primary goal of the UMG wrapper is to generate synthetic PA images corresponding to unknown PA materials, by transferring the style (texture) characteristics between fingerprint images of known PA materials. Gatys et al. [54] were the first to show that deep neural networks (DNNs) could encode not only content but also the style information. They proposed an optimization-based style-transfer approach, although prohibitively slow, for arbitrary images. In [158], Ulyanov et al. proposed use of an InstanceNorm layer to normalize feature statistics across spatial dimensions. An InstanceNorm layer is designed to perform the following operation:

$$IN(x) = \gamma\Big(\frac{x - \mu(x)}{\sigma(x)}\Big) + \beta \tag{3.4.1}$$

where, x is the input feature space, $\mu(x)$ and $\sigma(x)$ are the mean and standard deviation parameters, respectively, computed across spatial dimensions independently for each channel and each sample. It was observed that changing the affine parameters γ and β (while keeping convolutional parameters fixed) leads to variations in the style of the image, and the affine parameters could be learned for each particular style. This motivated an approach for artistic style transfer [38], which

learns γ and β values for each feature space and style pair. However, this required retraining of the network for each new style.

Huang and Belongie [73] replaced the InstanceNorm layer with an Adaptive Instance Norm (AdaIN) layer, which can directly compute affine parameters from the style image, instead of learning them – effectively transferring style by imparting second-order statistics from the target style image to the source content image, through the affine parameters. We follow the same approach as described in [73] in UMG wrapper for fusing feature statistics of one known (source) PA material image (c) providing friction ridge (content) information and source style, with another known, but different (target style) PA material (s) in the feature space. As described in AdaIN, we apply instance normalization on the input source image feature space, however, not with learnable affine parameters. The channel-wise mean and variance of the source image's feature space is aligned to match those of the target image's feature space. This is done by computing the affine parameters from the target material PA feature space in the following manner:

$$AdaIN(x, y) = \sigma(y)\Big(\frac{x - \mu(x)}{\sigma(x)}\Big) + \mu(y) \tag{3.4.2}$$

where the source (c) feature space is x and the target (s) feature space is y. In this manner, x is normalized with $\sigma(y)$ and shifted by $\mu(y)$. Our synthetic PA generator G is composed of an encoder $f(\cdot)$ and a decoder $g(\cdot)$. For the encoder, $f(\cdot)$, we use the first few layers of a pre-trained VGG-19 network similar to [88]. The weights of this network are frozen throughout all stages of the setup. For source image (c) and the target image (s), x is $f(c)$ and y is $f(s)$. The desired feature space is obtained as:

$$t = AdaIN(f(c), f(s)) \tag{3.4.3}$$

We use the decoder, $g(\cdot)$, to take t as input to produce $T(c, s) = g(t)$ which is the final synthesized image conditioned on the style from the target image. In order to ensure that our synthesized PA patches (*i.e.*, $g(t)$) do match the style statistics of the target material PA, we apply a style loss

L_s similar to [88, 98] given as:

$$\begin{aligned}\mathcal{L}_s = \textstyle\sum_{i=1}^{L} \left\| \mu\left(\phi_i(g(t))\right) - \mu\left(\phi_i(s)\right) \right\|_2 + \\ \textstyle\sum_{i=1}^{L} \left\| \sigma\left(\phi_i(g(t))\right) - \sigma\left(\phi_i(s)\right) \right\|_2\end{aligned} \tag{3.4.4}$$

where each ϕ_i denotes a layer in the encoder network (VGG-19). We pass $g(t)$ and s through $f(\cdot)$ and extract the outputs of $relu1\_1$, $relu2\_1$, $relu3\_1$ and $relu4\_1$ layers for computing $\mathcal{L}_s$.

The extent of style transfer can be controlled by interpolating between feature maps using:

$$T(c, s, \alpha) = g((1 - \alpha) \cdot f(c) + \alpha \cdot t) \tag{3.4.5}$$

where setting $\alpha = 0$ will reconstruct the original content image and $\alpha = 1$ will construct the most stylized image. To combine the two known styles, we preserve the style of source PA material while conditioning it with target PA material by setting the value of α to 0.5.

To ensure that the synthesized images retain friction ridge (fingerprint) content from the real image, we use a content loss, $\mathcal{L}_c$, which is computed as the euclidean distance between the features of the synthesized image, *i.e.*, $f(g(t))$ and the target features (t) from the real image.

$$\mathcal{L}_c = \left\| f(g(t)) - t \right\|_2 \tag{3.4.6}$$

Doing the style transfer, simply using a content loss ($\mathcal{L}_c$) to ensure that content is retained is not enough to ensure that the synthesized images look like real fingerprint images. Fingerprints have many details in terms of structure due to the presence of certain landmarks, *e.g.*, minutiae, ridges, and pores. With the aim of synthesizing fingerprints that look indistinguishable from the real fingerprints, we use adversarial supervision. A typical generative adversarial network (GAN) setup consists of a generator G and a discriminator D playing a *minimax game*, where D tries to distinguish between synthesized and real images, and G tries to fool D by generating realistic looking images. The adversarial objective functions for the generator ($\mathcal{L}_{adv}^{G}$) and discriminator

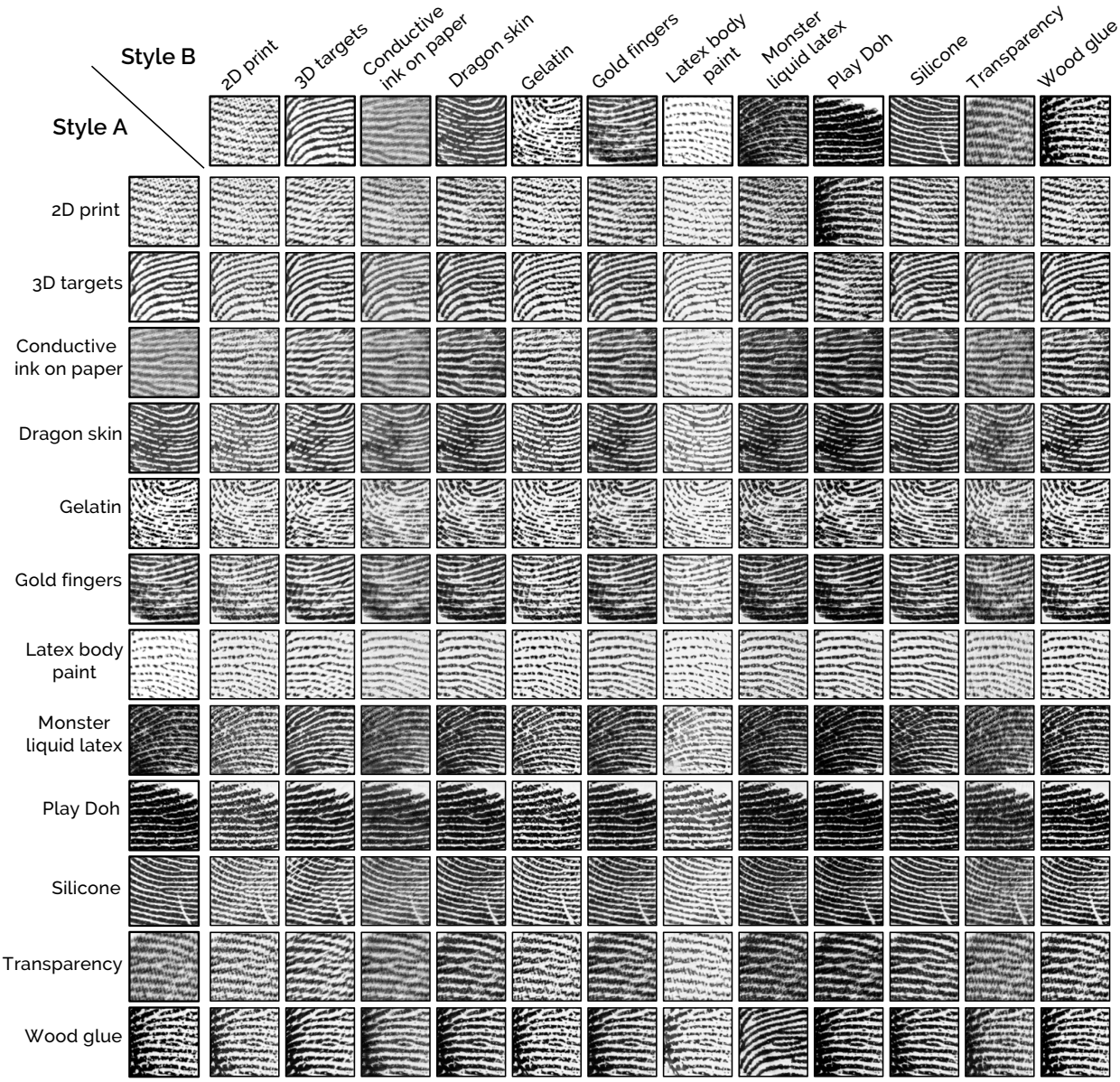


Figure 3.10 Synthesized PA patches (96×96) by the proposed Universal Material Generator using patches of a known (source) material (first column) conditioned on style ($\alpha = 0.5$) of another (target) known material (first row).

($\mathcal{L}_{adv}^{D}$) are given as[11]:

$$\mathcal{L}_{adv}^{G} = \mathbb{E}_t[\log(1 - D(G(t)))] \tag{3.4.7}$$

$$\mathcal{L}_{adv}^{D} = \mathbb{E}_x[\log D(x)] + \mathbb{E}_t[\log(1 - D(G(t)))] \tag{3.4.8}$$

In our approach, we use a discriminator as used in [133] and the generator is the decoder function $g(\cdot)$. We optimize the UMG wrapper in an end-to-end manner with the following objective functions:

$$\min_{G} \mathcal{L}_G = \lambda_c \cdot \mathcal{L}_c + \lambda_s \cdot \mathcal{L}_s + \mathcal{L}_{adv}^{G} \tag{3.4.9}$$

$$\max_{D} \mathcal{L}_D = \mathcal{L}_{adv}^{D} \tag{3.4.10}$$

where λ_c and λ_s are the weight parameters for content loss ($\mathcal{L}_c$) and style loss ($\mathcal{L}_s$), respectively. Algorithm 1 summarizes the steps involved in training a UMG wrapper.

3.4.1.3 UMG-Wrapper for PAD Generalization

Given a PA dataset of real images, S_{real}^{m}, fabricated using a set of m PA materials, we adopt a leave-one-out protocol to split the dataset such that PA images fabricated using $m-1$ materials are considered as "known" and used for training. And the images fabricated using the left-out m^{th} material are considered as "unknown" and used for computing the generalization performance. The fingerprint images of known materials ($k = m-1$) are used to train the UMG wrapper (UMG$_{spoof}$) described in section 3.4.1.2.

After we train the UMG$_{spoof}$, we utilize a total of N_{synth} randomly selected pairs of images $\{I_{m_a}^{i},\ I_{m_b}^{i}\}$ s.t. $i \in \{1, ..., N_{synth}\}$ from known but different materials $m_a, m_b \in \{m_1, ...m_k\}$, $a \neq b$, to generate a dataset of synthesized PA images S_{synth}^{k}. For each synthesized image, the friction ridge (content) information and the source material (style) characteristics are provided by the first image, I_{m_a}, and the target material (style) characteristics are provided by the second image,

[11]Here x is an image sampled from the distribution of real fingerprints, and t is the feature output by the AdaIN module.

Algorithm 1 Training UMG wrapper

procedure
input
    x: source image providing friction ridge content and known style A
    y: target image providing known style B
    $f(\cdot)$: encoder network; first 4 layers of VGG-19 network pre-trained on ImageNet with weights frozen during training
    $g(\cdot)$: decoder network; mirrors $f(\cdot)$ with pooling layers replaced with nearest up-sampling layers
    $D(\cdot)$: discriminator function similar to [133]
    $A(x, y)$: AdaIN operation; transfer style from x to y (using Eq. 3.4.2)
    $\alpha = 0.5$
    $\lambda_c = 0.001$, $\lambda_s = 0.002$
output
    $UMG(\cdot)$: UMG wrapper trained on known materials
begin:
    Encoding: $f_x = f(x)$ and $f_y = f(y)$
    Style transfer: $t = A(f_x, f_y)$
    Stylized image: $T(c, s, \alpha) = g((1 - \alpha) \cdot f_c + \alpha \cdot t)$
    Style Loss: $\mathcal{L}_s$ using Eq. 3.4.4
    Content Loss: $\mathcal{L}_c$ using Eq. 3.4.6
    Adversarial Loss (generator): $\mathcal{L}^G_{adv}$ using Eq. 3.4.7
    Adversarial Loss (discriminator): $\mathcal{L}^D_{adv}$ using Eq. 3.4.8
    Objective functions for training UMG wrapper
    $\min_G \mathcal{L}_G = \lambda_c \cdot \mathcal{L}_c + \lambda_s \cdot \mathcal{L}_s + \mathcal{L}^G_{adv}$
    $\max_D \mathcal{L}_D = \mathcal{L}^D_{adv}$
end

I_{m_b}. See Figures 3.9 and 3.10. The real PA dataset is augmented with the synthesized PA data to create a dataset that is used for training the fingerprint PA detector. Additionally, we augment the real live dataset with a total of N_{synth} synthesized live images using another UMG wrapper (UMG$_{live}$) trained on only live images. Adding synthesized live data balances the data distribution and forces the PA detector to learn generative-noise invariant features to distinguish between lives and PAs. Figure 3.11 presents examples of the synthesized live images.

The proposed Universal Material Generator approach acts like a wrapper on top of any existing PA detector to make it more robust to PAs not seen during training. In this study, we utilize two state-of-the-art spoof detectors, namely, Fingerprint Spoof Buster [24] and Slim-ResCNN [172]. Fingerprint Spoof Buster utilizes local patches (96×96) centered and aligned around fingerprint

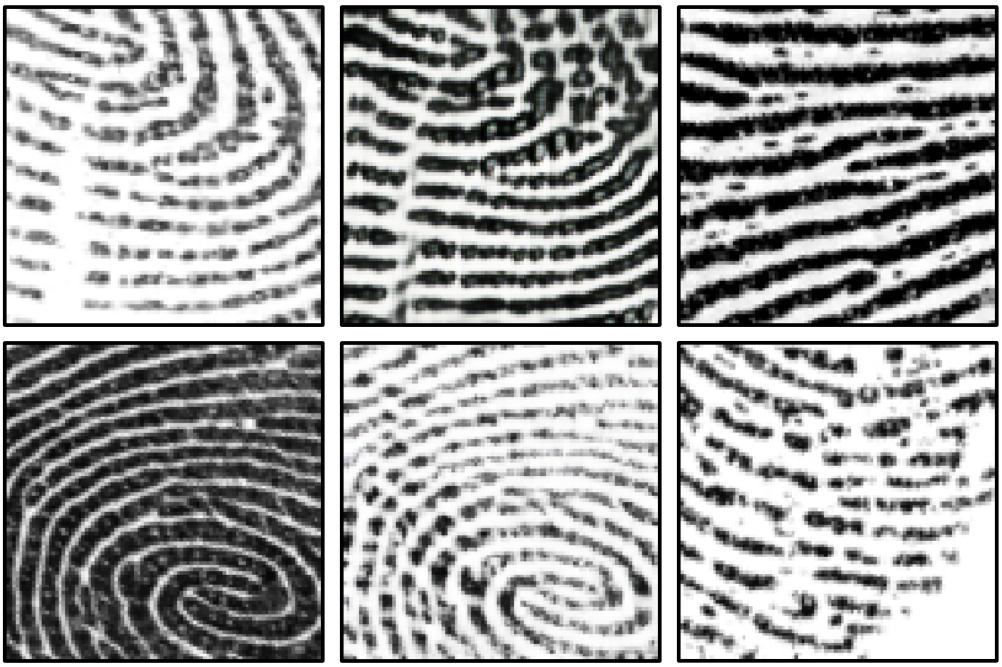


Figure 3.11 Synthetic live images generated by the proposed Universal Material Generator. (a) Source style images, (c) target style images, and (b) synthesized live images.

minutiae to train MobileNet-v1 [71] architecture and achieved state-of-the-art performance on publicly available LivDet databases [168] and exceeded the IARPA Odin Project [123] requirement of True Detection Rate (TDR) of 97.0% @ False Detection Rate (FDR) = 0.2%. Slim-ResCNN utilizes center of gravity-based local patches to train a custom CNN architecture containing residual blocks inspired from ResNet architecture [64], and achieved the best performance in the LivDet 2017 competition [114].

3.4.1.4 Experiments and Results

Minutiae Detection and Patch Extraction

The proposed UMG wrapper is trained on local patches of size 96×96 centered and aligned using minutiae points. We extract fingerprint minutiae using the algorithm proposed in [17]. For a given fingerprint image I with k detected minutiae points, $M = \{m_1, m_2, \ldots, m_k\}$, where $m_i =$

$\{x_i, y_i, \theta_i\}$, *i.e.*, the minutiae m_i is defined in terms of spatial coordinates (x_i, y_i) and orientation (θ_i), a corresponding set of k local patches $L = \{l_1, l_2, \ldots, l_k\}$, each of size $[96 \times 96]$, centered and aligned using minutiae location (x_i, y_i) and orientation (θ_i), are extracted as proposed in [24].

Implementation Details

The encoder of the UMG wrapper is the first four convolutional layers ($conv1\_1$, $conv2\_1$, $conv3\_1$, and $conv4\_1$) of a VGG-19 network [147] as discussed in section 3.4.1.2. We use encoder weights pre-trained on ImageNet [140] database which are frozen during training of the UMG wrapper. The decoder mirrors the encoder with pooling layers replaced with nearest up-sampling layers, and without use of any normalization layers as suggested in [73]. Both encoder and decoder utilize reflection padding to avoid border artifacts. The discriminator for computing the adversarial loss is similar to the one used in [133]. The weights for style loss and content loss are set to $\lambda_s = 0.002$ and $\lambda_c = 0.001$. We use the Adam optimizer [90] with a batch size of 8 and a learning rate of $1e-4$ for both generator (decoder) and discriminator objective functions. The input local patches are resized from 96×96 to 256×256 as required by the pre-trained encoder based on VGG-19 network. All experiments are performed in the TensorFlow framework.

The proposed approach is shown to improve the generalization performance of two state-of-the-art spoof detectors, namely, Fingerprint Spoof Buster and Slim-ResCNN. We train a MobileNet-V1 [71] classifier from scratch using the augmented dataset for Fingerprint Spoof Buster [24]. In the case of Slim-ResCNN, a custom architecture, consisting a series of optimized residual blocks [64] is implemented[12] as described in [172].

Experimental Protocol

The fingerprint PA generalization performance against unknown materials is evaluated by adopting a leave-one-out protocol [26]. In the case of MSU FPAD v2.0 dataset, one out of the twelve known PA materials is left-out and the remaining eleven materials are used to train the proposed UMG

[12]We were unable to obtain the source code for the Slim-ResCNN approach from the authors.

Figure 3.12 Example fingerprint images from LivDet 2017 database captured using three different fingerprint readers, namely Digital Persona, Green Bit, and Orcanthus. The unique characteristics of fingerprints from Orcanthus reader explain the performance drop in cross-sensor scenario when Orcanthus is used as either the source or the target sensor.

wrapper. The real PA data (of eleven known materials) is augmented with the synthesized PA data generated using the trained UMG wrapper, which is then used to train the fingerprint PA detector, *i.e.*, Fingerprint Spoof Buster [24]. This requires training a total of twelve different UMG wrappers and PA detection models each time leaving out one of the twelve different PA materials. The $5,743$ live images in MSUFPAD v2.0 are partitioned into training and testing such that there are $1,000$ randomly selected live images in testing set and the remaining $4,743$ images in training such that there is no subject overlap between training and testing data splits. The real live data is also augmented with synthesized live data generated using another UMG wrapper trained on real live data.

In the case of LivDet 2017 dataset, the PA materials available in the test set (Gelatin, Latex, and Liquid Ecoflex) are deemed as "unknown" materials because these are different from the materials included in the training set (Wood Glue, Ecoflex, and Body Double). To evaluate the generalization performance, we evaluate the performance of Fingerprint Spoof Buster with and without using the UMG wrapper and compare with the state-of-the-art published results. As the LivDet 2017 dataset contains fingerprint images from three different readers, we train two UMG wrappers per sensor, one for each of the live and the PA training datasets.

Table 3.5 Generalization performance (TDR (%) @ FDR = 0.2%) of state-of-the-art spoof detectors, *i.e.*, Slim-ResCNN [172] and Fingerprint Spoof Buster (FSB) [24], with leave-one-out method on MSU-FPAD v2 dataset. A total of twelve experiments are performed where the material left-out from training is taken as the "unknown" material for evaluation.

Unknown Spoof Material	**# Images**	**# Local Patches**	**Generalization Performance (TDR (%) @ FDR = 0.2%)**			
			Base CNN		Base CNN + UMG wrapper	
			Slim-ResCNN [172]	**Fingerprint Spoof Buster (FSB) [26]**	**Slim-ResCNN + UMG**	**FSB + UMG**
Silicone	$1,160$	$38,145$	64.74	67.59	**96.55**	**98.62**
Monster Liquid Latex	882	$27,458$	90.25	94.78	**95.35**	**96.26**
Play Doh	715	$17,602$	58.18	58.46	**71.05**	**72.31**
2D Printed Paper	481	$7,381$	53.22	55.30	**79.42**	**80.25**
Wood Glue	397	$12,681$	84.89	86.40	**97.98**	**98.99**
Gold Fingers	295	$9,402$	85.08	88.14	**88.14**	**88.81**
Gelatin	294	$10,508$	55.78	55.10	**98.30**	**97.96**
Dragon Skin	285	$7,700$	96.14	97.54	**99.30**	**100.00**
Latex Body Paint	176	$6,366$	78.98	76.70	**90.34**	**89.20**
Transparency	137	$3,846$	91.24	95.62	**97.08**	**100.00**
Conductive Ink on Paper	50	$2,205$	88.00	90.00	**96.00**	**100.00**
3D Universal Targets	40	$1,085$	92.50	95.00	**100.00**	**100.00**
Total Spoofs	**4,912**	**144,379**	**Weighted mean* ($\pm$ weighted s.d.)**			
Total Lives	**5,743**	**228,143**	**73.09 $\pm$ 15.66**	**75.24 $\pm$ 16.60**	**90.63 $\pm$ 10.19**	**91.78 $\pm$ 10.29**

*The generalization performance for each spoof material is weighted by the number of images to produce the weighted mean and standard deviation.

Cross-Material Fingerprint PA Generalization

Table 3.5 presents the generalization performance of the proposed approach on the MSU FPAD v2.0 dataset. The mean generalization performance of the spoof detector against unknown spoof materials improves from TDR of 75.24% (73.09%) to TDR of 91.78% (90.63%) @ FDR = 0.2% for Fingerprint Spoof Buster (Slim-ResCNN), resulting in approximately 67% decrease in the error rate, when the spoof detector is trained in conjunction with the proposed UMG wrapper. Table 3.6 presents a performance comparison of the proposed approach and the state-of-the-art approach when tested on the publicly available LivDet 2017 dataset. The proposed UMG wrapper improves the state-of-the-art average cross-material spoof detection performance from TDR = 73.32% (72.62%) to 80.74% (78.27%) @ FDR = 1.0% for Fingerprint Spoof Buster (Slim-ResCNN), respectively.

Table 3.6 Performance comparison between the proposed approach and state-of-the-art CNN-only results [24, 172] on LivDet 2017 dataset for cross-material experiments in terms of Average Classification Accuracy (ACA) and TDR @ FDR = 1.0%.

LivDet 2017	**Base CNN**		**Base CNN + UMG wrapper**	
	Slim-ResCNN* [172]	**FSB [26]**	**Slim-ResCNN + UMG**	**FSB + UMG**
	Avg. Accuracy (TDR @ FDR = 1.0%)		**Avg. Accuracy (TDR @ FDR = 1.0%)**	
Green Bit	95.20 (90.22)	96.68 (91.07)	96.90 (91.95)	97.42 (92.29)
Orcanthus	93.93 (65.82)	94.51 (66.59)	94.45 (71.91)	95.01 (74.45)
Digital Persona	92.89 (61.81)	95.12 (62.29)	94.75 (70.96)	95.20 (75.47)
Mean ± s.d.	94.01 ± 1.16 (72.62 ± 15.38)	95.44 ± 1.12 (73.32 ± 15.52)	**95.37 ± 1.34** (**78.27 ± 11.85**)	**95.88 ± 1.34** (**80.74 ± 10.02**)

*We were unable to obtain the source code for the Slim-ResCNN approach from the authors. Best efforts were made to implement the approach based on the details provided in their manuscript [172]. Based on LivDet 2017 [114], Slim-ResCNN achieved average classification accuracy of 95.25% compared to 94.01% achieved by our implementation.

Table 3.7 Cross-sensor fingerprint spoof generalization performance on LivDet 2017 dataset in terms of Average Classification Accuracy and TDR @ FDR = 1.0%.

LivDet 2017	**Slim-ResCNN [172]**	**FSB [26]**	**Slim-ResCNN + UMG**	**FSB + UMG**
Training (Testing) Sensors	**Avg. Accuracy (TDR @ FDR = 1.0%)**		**Avg. Accuracy (TDR @ FDR = 1.0%)**	
Green Bit (Orcanthus)	43.98 (0.00)	49.43 (0.00)	**65.40** (**20.60**)	**66.05** (**21.52**)
Green Bit (Digital Persona)	80.39 (48.28)	89.37 (57.48)	**92.07** (**69.55**)	**94.81** (**72.91**)
Orcanthus (GreenBit)	68.82 (8.02)	69.93 (8.02)	**74.38** (**29.90**)	**81.75** (**30.91**)
Orcanthus (Digital Persona)	62.30 (6.70)	57.99 (4.97)	**72.33** (**25.24**)	**76.36** (**28.46**)
Digital Persona (GreenBit)	87.90 (54.24)	89.54 (57.06)	**95.28** (**84.38**)	**96.35** (**85.21**)
Digital Persona (Orcanthus)	44.30 (0.00)	49.32 (0.00)	**66.10** (**18.25**)	**68.44** (**20.38**)
Mean ± s.d.	64.62 ± 18.18 (19.54 ± 24.86)	67.60 ± 18.53 (21.26 ± 28.06)	**77.59 ± 12.97** (**41.32 ± 28.29**)	**80.63 ± 12.88** (**43.23 ± 28.31**)

Cross-Sensor Fingerprint PA Generalization

To improve the cross-sensor performance, we employ the proposed UMG wrapper to synthetically generate large-scale live and PA datasets to train a PA detector for the target sensor. Given a real fingerprint database, D^A_{real}, collected on a source fingerprint sensor, F^A, containing real live, L^A_{real}, and real PA, S^A_{real} datasets, s.t. $D^A_{real} = \{L^A_{real} \cup S^A_{real}\}$, the proposed UMG wrapper is used to generate $50,000$ synthetic live patches, L^B_{synth}, and $50,000$ synthetic PA patches, S^B_{synth}, for a target sensor, F_B. The UMG wrapper is trained only on the live images collected on S_B, and used for style transfer on L^A_{real} and S^A_{real} to generate L^B_{synth}, and S^B_{synth}, respectively. We evaluate the cross-sensor generalization performance using LivDet 2017 dataset where the UMG wrapper trained on a source sensor, say Green Bit, is used to generate synthetic data for a target sensor, say Orcanthus,

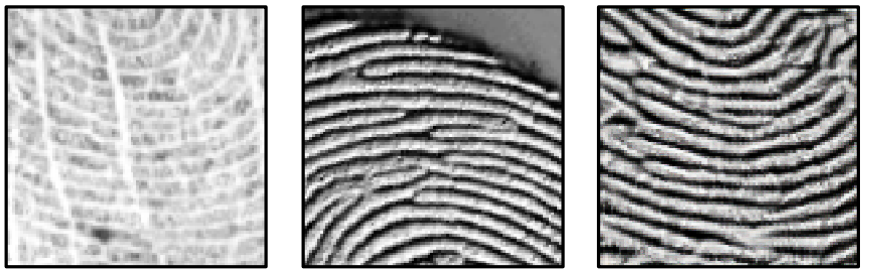


Figure 3.13 UMG wrapper used to transfer style from (b) a real live patch from Orcanthus reader, to (a) a real live patch from Digital Persona, to generate (c) a synthesized patch.

using only a small set of 100 live fingerprint images from the target sensor[13]. The PA detector is trained from scratch only on the synthetic dataset created for the target sensor using UMG wrapper and tested on the real test set of the target sensor. Table 3.7 presents the cross-sensor fingerprint PA generalization performance of the PA detector in terms of average classification accuracy and TDR (%) @ FDR = 1%. We note that the proposed UMG wrapper improves the average cross-sensor PA detection performance from 67.60% to 80.63%. Figure 3.12 presents example fingerprint images captured using the three sensors in LivDet 2017. The unique characteristics of fingerprints from Orcanthus reader explain the performance drop in cross-sensor scenario when it is used as either the source or the target sensor.

3.4.1.5 Computational Requirements

Offline Training stage: The proposed approach includes an offline stage of training the UMG wrapper and synthesis of style-transferred fingerprint patches. It takes around 2 hours to train, and around 1 hour to generate $100,000$ fingerprint patches on a Nvidia GTX 1080Ti GPU. The synthesized fingerprint patches are used to augment the training data used to train the underlying spoof detector.

Online Testing stage: There is no increase in the spoof detection time of the underlying spoof detector with the addition of the UMG wrapper. The spoof detection time remains around 100ms for both Fingerprint Spoof Buster and Slim-ResCNN.

[13] An average of ~ 3100 local patches are extracted from 100 live fingerprint images in LivDet 2017 experiments.

(a) Real spoof A (Silicone) (b) Real spoof B (Latex Body Paint) (c) Real spoof mixture (spoof A + spoof B) (d) Synthesized spoof (spoof A + spoof B)

Figure 3.14 Fingerprint patches fabricated with real PAs (a) silicone, (b) latex body paint, (c) their mixture (in 1:1 ratio), and (d) synthesized using UMG wrapper with style transfer between silicone and latex body paint.

3.4.1.6 Fabricating Unknown PAs

To explore the role of cross-material style transfer in improving generalization performance, we fabricate physical PA specimens using two PA materials, namely silicone and latex body paint, and their mixture in a 1:1 ratio by volume[14]. We fabricate a total of 24 physical specimens, including 8 specimens for each of the two materials, and 8 specimens using their mixture. A total of 72 PA fingerprints, 3 impressions/specimen, are captured using a CrossMatch Guardian 200 fingerprint reader. Fingerprint Spoof Buster, trained on twelve known PA materials including silicone and latex body paint, achieves TDR of 100% @ FDR = 0.2% on the two known PA materials, and TDR of 83.33% @ FDR = 0.2% against the mixture. We utilize the testing dataset of $1,000$ live fingerprint images from MSU FPAD v2.0 for these experiments.

We utilize the proposed UMG wrapper to generate a dataset of $5,000$ synthesized PA patches[15] using cross-material style transfer between PA fingerprints of silicone and latex body paint. Fingerprint Spoof Buster, fine-tuned using the synthesized dataset, improves the TDR from 83.33% to 95.83% @ FDR = 0.2% when tested on the silicone and latex body paint mixture, highlighting the role of the style-transferred synthesized data in improving generalization performance. Fig-

[14]Not all PA materials can be physically combined and may result in mixtures with poor physical properties for them to be used to fabricate any good quality PA artefacts.

[15]Around 1,100 minutiae-based local patches are extracted from 24 fingerprint images corresponding to each material.

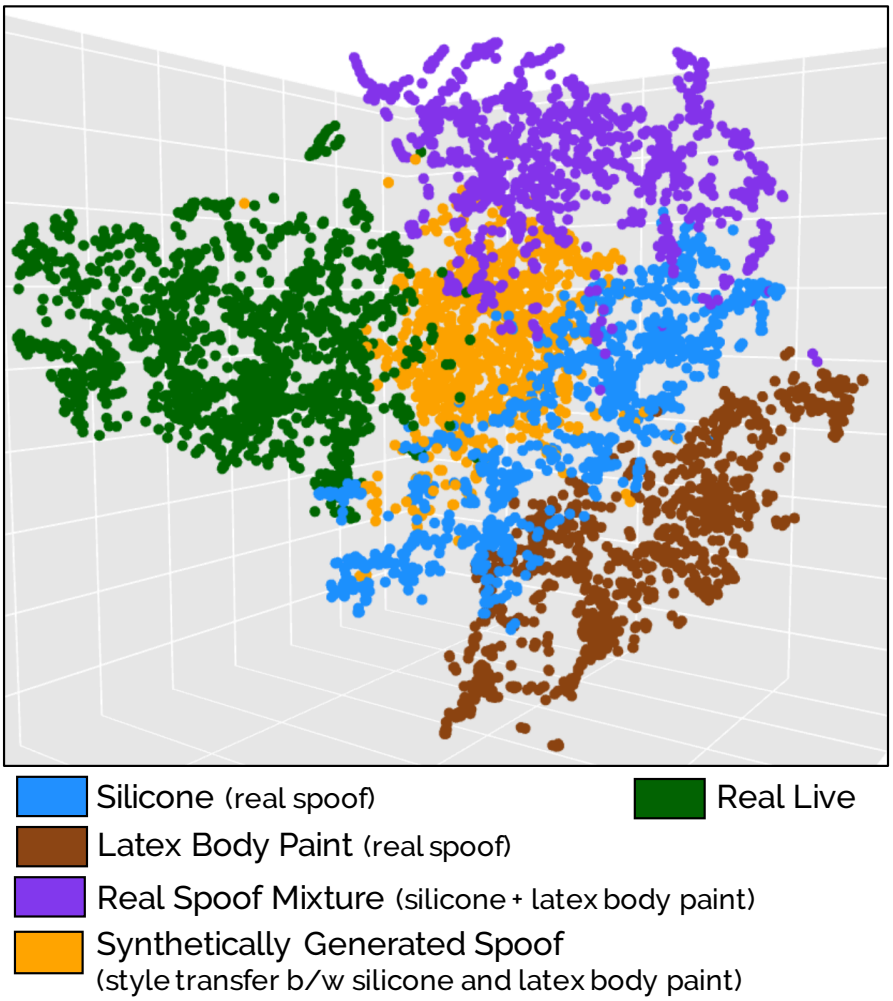


Figure 3.15 3D t-SNE visualization of feature embeddings of real live fingerprints, PA fingerprints fabricated using silicone, latex body paint, and their mixture (1:1 ratio), and synthesized PA fingerprints using style-transfer between silicone and latex body paint PA fingerprints. The 3D embeddings are available at http://tarangchugh.me/posts/umg/index.html (Best viewed in color)

ure 3.14 presents sample fingerprint patches of the two PA materials, silicone and latex body paint, their physical mixture, and synthesized using style-transfer. Figure 3.15 presents the 3D t-SNE visualization of feature embeddings of live fingerprints (green), two materials, silicone (blue) and latex body paint (brown), their mixture (purple), and synthetically generated images (orange). Although the mixture embeddings are not located in between the embeddings for the two known materials, possibly due to the low-dimensional t-SNE representation, they are close to the embeddings of the synthetically generated PA images. This explains the improvement in performance

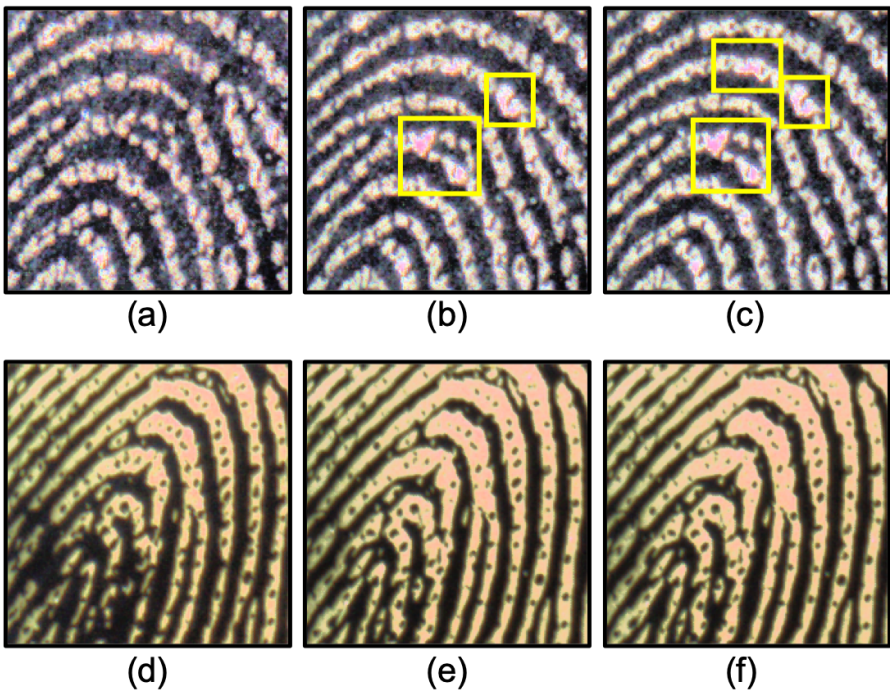


Figure 3.16 A sequence of ten color frames are captured by a SilkID SLK20R fingerprint reader in quick succession (8 fps). The first, fifth, and tenth frames from a live (a) - (c), and PA (tan pigmented third degree) (d) - (f) finger are shown here. Unlike PAs, in the case of live fingers, appearance of sweat near pores (highlighted in yellow boxes) and changes in skin color (pinkish red to pale yellow) along the frames can be observed.

against the PA mixtures when synthesized PAs are used in training. Therefore, the proposed UMG wrapper is able to generate PA images that are potentially similar to the unknown PAs.

3.4.2 Temporal Analysis for PAD Generalization

In this section, we present a dynamic approach to improve the PAD generalization [27]. We propose to utilize the dynamics involved in the imaging of a fingerprint on a touch-based fingerprint reader, such as perspiration, changes in skin color (blanching), and skin distortion, to differentiate bonafide fingers from PA fingers. Specifically, we utilize a deep learning-based architecture (CNN-LSTM) trained end-to-end using sequences of minutiae-centered local patches extracted from ten color frames captured on a COTS fingerprint reader (SilkID Fast Frame Rate sensor).

Compared to the static approaches that were discussed earlier, in the case of dynamic approaches, published studies utilize temporal analysis to capture the physiological features, such

Table 3.8 Studies primarily focused on fingerprint presentation attack detection using temporal analysis.

Study	Approach	Database	Performance
Parthasaradhi et al. [128]	Temporal analysis of perspiration pattern along friction ridges	1,840 live from 33 subjects and 1800 PA from 2 materials, and 700 cadaver from 14 fingers	Avg. Classification Accuracy = 90%
Kolberg et al. [91]	Blood flow detection using a sequence of 40 Laser Speckle Contrast Images	1,635 live from 163 subjects and 675 PA images of 8 PA materials (32 variants)	TDR = 90.99% @ FDR = 0.05%
Plesh et al. [131]	Fusion of static (LBP and CNN) and dynamic (changes in color ratio) features using a sequence of 2 color frames	14,892 live and 21,700 PA images of 10 materials	TDR = 96.45% (known-material) @ FDR = 0.2%
Proposed Approach	Temporal analysis of minutiae-based local patch sequences from 10 color frames using CNN + LSTM model	26,650 live from 685 subjects and 32,910 PA images of 7 materials (14 variants)	TDR = **99.15%** (known-material) and TDR = **86.20%** (cross-material) @ FDR = 0.2%

as perspiration [106, 128], blood flow [91, 169], skin distortion [2], and color change [131, 169]. Table 4.1 summarizes the dynamic approaches for fingerprint PA detection reported in the literature. Some of the limitations of these studies include long capture time (2-5 seconds), expensive hardware, and/or small number of frames in the sequence. Moreover, it is likely that some live fingers may not exhibit any of these dynamic phenomenons to separate them from PAs. For instance, some dry fingers may not exhibit signs of perspiration during the finger presentation or a PA may produce similar distortion characteristics as that of some live fingers.

We posit that automatic learning, as opposed to hand-engineering, of the dynamic features involved in the presentation of a finger can provide more robust and highly discriminating cues to distinguish live fingerprints from PAs. In this section, we propose to use a CNN-LSTM architecture to learn the spatio-temporal features across different frames in a sequence. We utilize a sequence of minutiae-centered local patches extracted from ten colored frames captured by a COTS fingerprint reader, SilkID SLK20R[16], at 8 fps to train the network in an end-to-end manner. The use of minutiae-based local patches has been shown to achieve state-of-the-art PA detection performance compared to randomly selected local patches in static images. Additionally, using minutiae-based

[16]https://www.zkteco.com/en/product_detail/SLK20R.html

local patches provides a large amount of training data, $71,530$ minutiae-based patch sequences, compared to $5,956$ whole-frame sequences.

3.4.2.1 Proposed Approach

The proposed approach consists of: (a) detecting minutiae from each of the frames and selecting the frame with the highest number of minutiae as the reference frame, (b) preprocessing the sequence of frames to convert them from Bayer pattern grayscale images to RGB images, (c) extracting local patches[17] from all ten frames based on the location of detected minutiae in the reference frame, and (c) end-to-end training of a CNN-LSTM architecture using the sequences of minutiae-centered patches extracted from the ten frames. While a time-distributed CNN network (MobileNet-v1) with shared weights extracts deep features from the local patches, a bidirectional LSTM layer is utilized to learn the temporal relationship between the features extracted from the sequence. An overview of the proposed approach is presented in Figure 3.19.

Minutia Detection

When a bonafide finger (or PA) is presented to the SilkID SLK20R fingerprint reader, it captures a sequence of ten color frames, $F = \{f_1, f_2, ..., f_{10}\}$, at 8 frames per second[18] (fps) and a resolution of 1000 ppi. While the complete sensing region ($h \times w$) in a SilkID fingerprint reader is 800×600 pixels, each of the ten colored frames are captured from a smaller central region of 630×390 pixels to ensure the fast frame rate of 8 fps. The starting and ending frames in the sequence may have little or no friction ridge details if the finger is not yet completely placed or quickly removed from the reader. Therefore, we extract minutiae information from all of the ten frames using the algorithm proposed by Cao et al. [17]. Since the minutiae detector proposed in [17] is optimized for 500 ppi fingerprint images, all frames are resized before extracting the minutiae. The frame

[17]Earlier, we reported that for 500 ppi fingerprint images, the minutiae-based patches of size 96×96 pixels achieve the best performance compared to other patch sizes. Since SilkID fingerprint images have a resolution of 1000 ppi, we select a patch size of 192×192 pixels to ensure a similar amount of friction ridge area in each patch, as contained in a 96×96 pixels patch size for 500 ppi fingerprint images.

[18]It takes an average of 1.25 seconds to capture a sequence of ten frames.

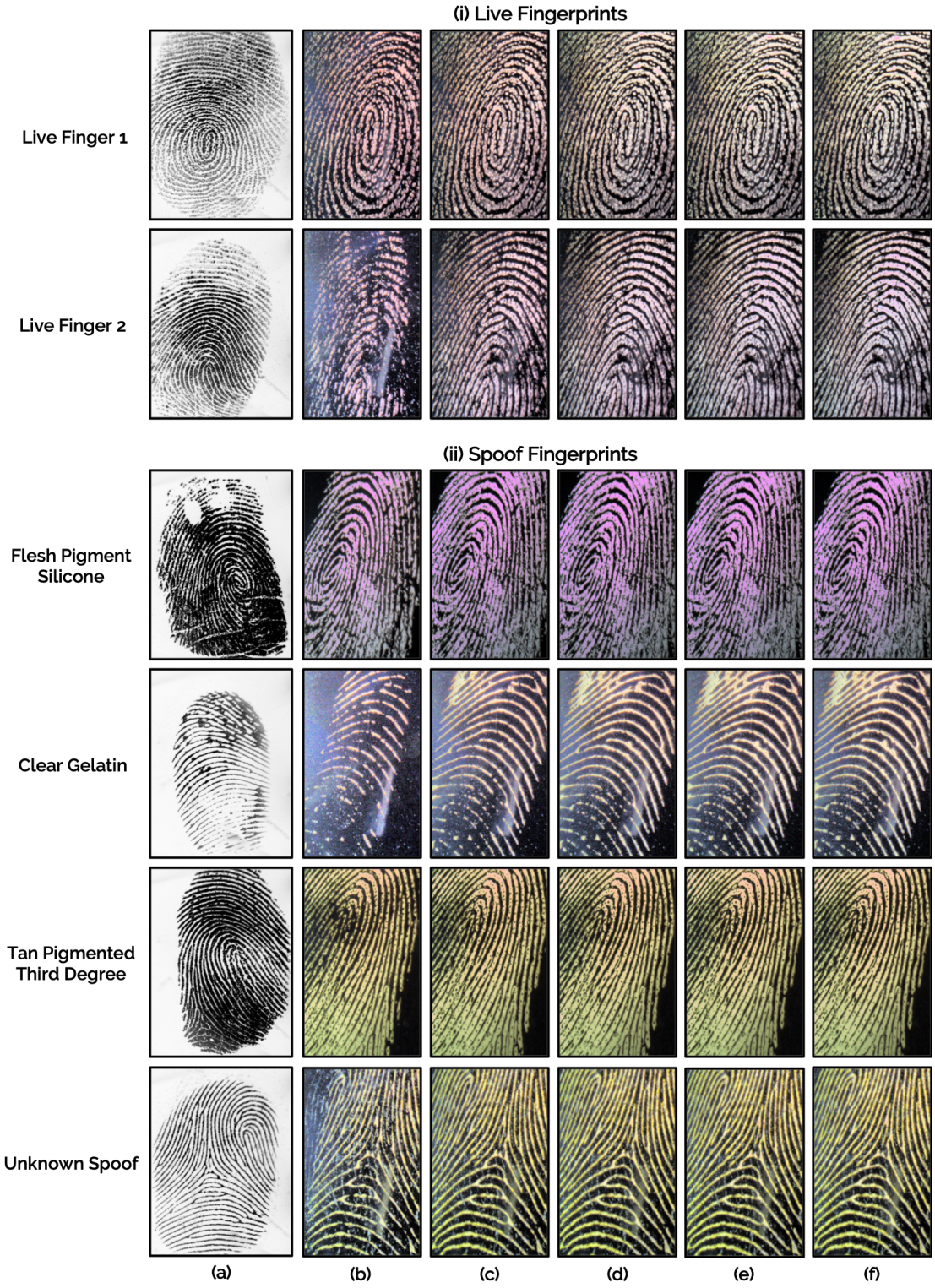


Figure 3.17 Examples of (i) live and (ii) PA fingerprint images. (a) Grayscale 1000 ppi image, and (c)-(g) the first five (colored) frames captured by SilkID SLK20R Fast Frame Rate reader. Live frames exhibit the phenomenon of blanching of the skin, *i.e.*, displacement of blood when a live finger is pressed on the glass platen changing the finger color from red/pink to pale white. (Best viewed in color)

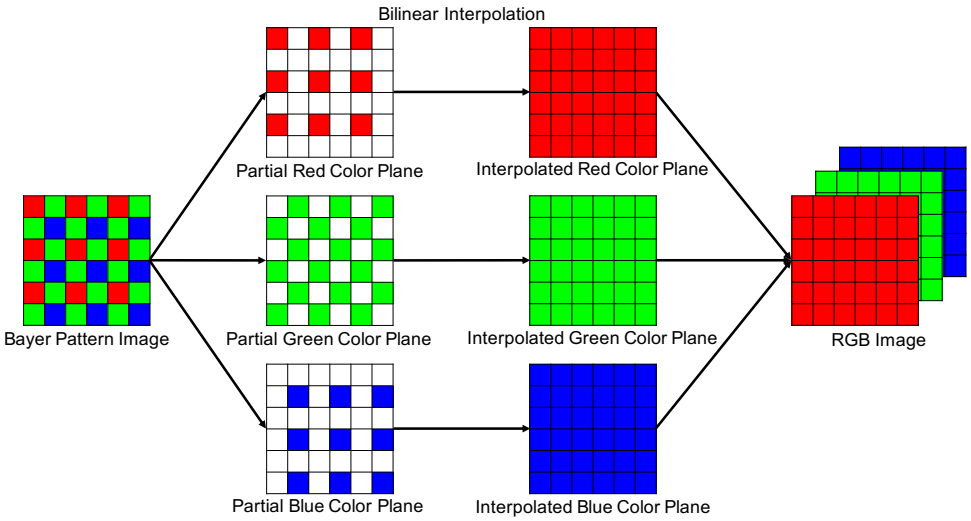


Figure 3.18 A Bayer color filter array consists of alternating rows of red-green and green-blue filters. Bilinear interpolation of each channel is utilized to construct the RGB image.

with the maximum number of detected minutiae is selected as the reference frame (f^{ref}) and the corresponding minutiae set as the reference minutiae set (M^{ref}).

Pre-processing

A digital sensor, containing a large array of photo-sensitive sites (pixels), is typically used in conjunction with a color filter array to permit only particular colors of light at each pixel. The SilkID fingerprint reader employs one of the most common filter arrays, known as *Bayer filter array*, consisting of alternating rows of red-green (RG) and green-blue (GB) filters. *Bayer demosaicing* [97] (debayering) is the process of converting a bayer pattern image to an image with complete RGB color information at each pixel. It utilizes bilinear interpolation [153] to estimate the missing pixels in the three color planes as shown in Figure 3.18. The original sequence of grayscale Bayer pattern frames ($10 \times 630 \times 390$) is converted to the RGB colorspace using an OpenCV [11] function, *cv2.cvtColor()*, with the parameter $flag =$ *cv2.COLOR_BAYER_BG2RGB*. After debayering, the frames have high pixel intensity values in the green channel (see Figure 3.19) as SilkID readers are calibrated with strong gains on green pixels for generating high quality FTIR images. We utilize these raw images for our experiments. For visualization purposes, we reduce the green channel

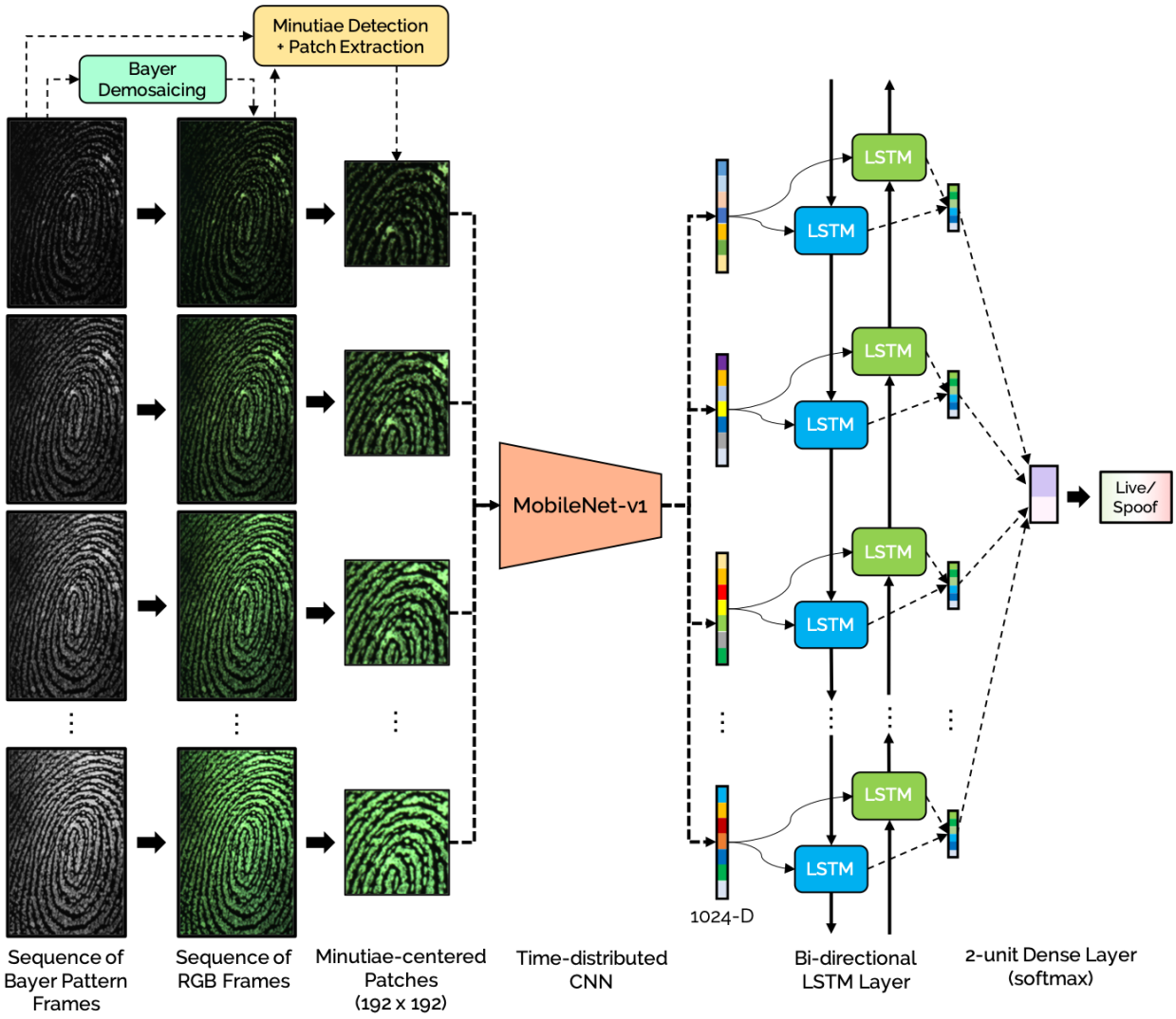


Figure 3.19 An overview of the proposed approach utilizing a CNN-LSTM model trained end-to-end on sequences of minutiae-centered local patches for fingerprint PA detection.

intensity values by a factor of 0.58 and perform histogram equalization on intensity values in the HSV colorspace[19] (see Figures 3.16 and 3.17).

Local Patch Extraction

For each of the detected minutiae from the reference frame, $m_i \in M^{ref}$, we extract a sequence of ten local patches, $P_i = \{p_i^{f_1}, p_i^{f_2}, ..., p_i^{f_{10}}\}$, of size 192×192, from the ten frames (F), centered

[19]Reducing gain in green channel and histogram equalization achieved similar or lower performance compared to using raw color images. Therefore, raw images were used for all experiments.

at the minutiae location[20], *i.e.*, $m_i = \{x_i, y_i\}$. This results in a total of k patch sequences, where k is equal to the number of detected minutiae in the reference frame. Earlier, we reported that for 500 ppi fingerprint images, the minutiae-based patches of size 96×96 pixels achieve the best performance compared to patch sizes of 64×64 pixels and 128×128 pixels. Therefore, for 1000 ppi images in our case, we selected the patch size of 192×192 pixels to ensure a similar amount of friction ridge area in each patch, as contained in a 96×96 pixels patch size for 500 ppi fingerprint images. Each local patch from the reference frame is centered around the minutiae. However, this might not hold true for non-reference frames where the minutiae may shift due to non-linear distortion of human skin and non-rigid PA materials. We hypothesize that the proposed approach can utilize the differences in the non-linear shift along the sequences of local patches as a salient cue to distinguish between live and PAs.

3.4.2.2 Network Architecture

Several deep Convolutional Neural Network (CNN) architectures, such as VGG [147], Inception-v3 [150], MobileNet-v1 [71] etc., have been shown to achieve state-of-the-art performance for many vision-based tasks, including fingerprint PA detection [23, 119]. Unlike traditional approaches where spatial filters are hand-engineered, CNNs can automatically learn salient features from the given image databases. However, as CNNs are feed-forward networks, they are not well-suited to capture the temporal dynamics involved in a sequence of images. On the other hand, a Recurrent Neural Network (RNN) architecture with feedback connections can process a sequence of data to learn the temporal features.

With the goal of learning highly discriminative and generalizable spatio-temporal features for fingerprint PA detection, we utilize a joint CNN-RNN architecture that can extract deep spatial features from each frame, and learn the temporal relationship across the sequence. One of the most popular RNN architectures is Long Short-Term Memory [70] that can learn long range dependencies from the input sequences. The proposed network architecture utilizes a time-distributed

[20]Minutiae coordinates extracted from the resized 500 ppi frames are doubled to correspond to minutiae coordinates in the original 1000 ppi frames.

MobileNet-v1 CNN architecture followed by a Bi-directional LSTM layer[21] and a 2-unit softmax layer for the binary classification problem, *i.e.*, live vs. PA. See Figure 3.19.

MobileNet-v1 is a low-latency network with only 4.24M trainable parameters compared to other networks, such as Inception-v3 (23.2M) and VGG (138M), which achieve comparable performance in large-scale vision tasks [140]. In low resource requirements such as smartphones and embedded devices, MobileNet-v1 is well-suited for real-time PA detection. Most importantly, it has been shown to achieve state-of-the-art performance for fingerprint PA detection [24] on publicly available datasets [57]. It takes an input image of size $224 \times 224 \times 3$, and outputs a 1024-dimensional feature vector (bottleneck layer). We resize the local patches from 192×192 to 224×224 as required by the MobileNet-v1 input. For the purposes of processing a sequence of images, we utilize a Keras' TimeDistributed wrapper to utilize the MobileNet-v1 architecture as a feature extractor with shared parameters across different frames (time-steps) in the sequence.

3.4.2.3 Implementation Details

The network architecture is designed in the Keras framework[22] and trained from scratch on a Nvidia GTX 1080Ti GPU. We utilize the MobileNet-v1 architecture without its last layer wrapped in a Time-Distributed layer. The Bi-directional LSTM layer contains 256 units and has a dropout rate of 0.25. We utilize the Adam [90] optimizer with a learning rate of 0.001 and a binary cross entropy loss function. The network is trained end-to-end with a batch size of 4. The network is trained for 80 epochs with early-stopping[23].

Table 3.9 Performance comparison (TDR (%) @ FDR = 0.2% and 1.0%) between the proposed approach and two state-of-the-art methods [24, 172] for known-material scenario, where the spoof materials used in testing are also known during training.

Study	Approach	Architecture	TDR ($\pm$ s.d.) (%) @ FDR = 0.2%	TDR ($\pm$ s.d.) (%) @ FDR = 1.0%
Baseline	Static (Whole Image)	CNN (MobileNet-v1)	96.90 ± 0.78	97.64 ± 0.55
Zhang et al. [172]	Static (Center of Gravity Patches)	CNN (Slim-ResCNN)	98.05 ± 0.38	98.44 ± 0.30
Chugh et al. [24]	Static (Minutiae Patches)	CNN (MobileNet-v1)	99.11 ± 0.24	99.15 ± 0.24
Proposed	Dynamic (Whole Frames)	CNN-LSTM (MobileNet-v1)	98.94 ± 0.44	99.04 ± 0.43
	Dynamic (Center of Gravity Patches)	CNN-LSTM (Slim-ResCNN)	99.04 ± 0.26	99.30 ± 0.28
	Dynamic (Minutiae Patches)	CNN-LSTM (MobileNet-v1)	**99.25 $\pm$ 0.22**	**99.45 $\pm$ 0.16**

Table 3.10 Performance comparison (TDR (%) @ FDR = 0.2% and 1.0%) between the proposed approach and two state-of-the-art methods [24, 172] for three cross-material scenarios, where the spoof materials used in testing are unknown during training.

	Baseline Static Approaches (CNN)			**Proposed Dynamic Approaches (CNN-LSTM)**		
Unknown Material	**Whole Image (Grayscale)**	**Slim-ResCNN [172]**	**Fingerprint Spoof Buster [24]**	**Sequence of Whole Images**	**Sequence of CoG Patches**	**Sequence of Minutiae-based Patches**
	TDR @ FDR = 0.2%					
Third Degree	43.83	75.32	79.20	80.44	83.22	84.50
Gelatin	50.74	76.84	76.52	73.88	83.10	82.81
Ecoflex	77.37	87.39	89.23	87.55	90.94	91.28
Mean $\pm$ s.d.	57.31 ± 17.71	79.85 ± 6.57	81.65 ± 6.70	80.62 ± 6.84	85.75 ± 4.49	**86.20 $\pm$ 4.48**
	TDR @ FDR = 1.0%					
Third Degree	60.25	86.15	89.11	88.10	94.22	96.20
Gelatin	66.40	90.10	89.00	89.50	96.38	96.08
Ecoflex	85.31	93.27	94.90	93.27	98.00	98.20
Mean $\pm$ s.d.	70.65 ± 13.06	89.84 ± 3.57	91.00 ± 3.37	90.29 ± 2.67	96.20 ± 1.90	**96.83 $\pm$ 1.19**

3.4.2.4 Experimental Results

To demonstrate the robustness of our proposed approach, we evaluate it using the SilkID Fast Frame Rate dataset (Table 3.3) under two different settings: *Known-Material* and *Cross-Material* scenarios.

Known-Material Scenario

In this scenario, the same set of PA materials are included in the train and test sets. To evaluate this, we utilize five-fold cross validation splitting the live and PA datasets for training and testing

[21] Experiments with uni-directional LSTM layer achieved lower or similar performance compared to when using bi-directional layer.

[22] https://keras.io/

[23] The patience parameter is set to 20, which means that if the validation accuracy does not improve for more than 20 epochs the network training is automatically stopped.

with no subject overlap. In each of the five folds, there are $21,320$ live and $26,400$ PA frames in training and the rest are in testing. Table 3.9 presents the results achieved by the proposed approach on known-materials compared to a state-of-the-art approach [24] that utilizes minutiae-based local patches from static grayscale images. The proposed approach improves the spoof detection performance from TDR of 99.11% (99.15%) to 99.25% (99.45%) @ FDR = 0.2% (1.0%).

Cross-Material Scenario

In this scenario, the PA materials used in the test set were not included in the training set. We simulate this scenario by adopting a leave-one-out protocol, where one material (including all its variants) is removed from training, and is then used for evaluating the trained model. It is a more challenging and practical setting as it evaluates the cross-material generalizability of a PA detector against PA materials that are never seen during training. For instance, in one of the cross-material experiments, we exclude Third Degree silicone PA material, including its all variants (pigmented, tan, beige powder, and medium) from training, and use them for testing. The live data is randomly divided in a 80/20 split, with no subject overlap, for training and testing, respectively.

Table 3.10 presents the performance achieved by the proposed approach, on three cross-material experiments, compared to two state-of-the-art[24] methods [24, 172]. We observe that utilizing sequence of whole images significantly improves the performance achieved by static whole images (from TDR = 57.31% (70.65%) to TDR = 80.62% (90.29%) @ FDR = 0.2% (1.0%)). However, it is slightly lower that the performance achieved by the static patch-based approaches, *i.e.*, TDR = 81.65% (91.00) @ FDR = 0.2% (1.0%). This could be due to the drawbacks of utilizing whole images compared to local patches [24] for training a deep neural network, namely, (i) whole images may have some blank area surrounding the friction ridge area; directly resizing these images, from 630×390 to 224×224, results in the friction ridge area occupying less than 20% of the original image size, (ii) resizing a rectangular image to a square image leads to different amounts of information retained in the two spatial dimensions, and (iii) downsizing an image typically leads to significant loss of discriminatory information. However, these drawbacks are addressed by using

[24]The algorithm by Zhang et al. [172], Slim-ResCNN, was the winner of the LivDet 2017 competition [114].

a sequence of local patches in the proposed approach, which is shown to achieve a superior cross-material PA detection performance of TDR s= 86.20% (96.83%) @ FDR = 0.2% (1.0%). Figure **??** presents three challenging cases in the case of cross-material experiment where the Third Degree silicone PA material is left out from training, and is used in testing.

3.4.2.5 Processing Times

The proposed network architecture takes around $4-6$ hours to converge when trained with sequences of whole frames, and $24-30$ hours with sequences of minutiae-based local patches, using a Nvidia GTX 1080Ti GPU. An average number of 11 and 13 sequences of minutiae-based local patches are extracted from the live and PA frames, respectively. The average classification time for a single presentation, including: preprocessing, minutiae-detection, patch extraction, and sequence generation and inference, on a Nvidia GTX 1080 Ti GPU, is 58ms for full frame-based sequences, and 393ms for minutiae-based patch sequences.

3.5 Summary

Introduction of new PA materials and fabrication techniques poses a continuous threat to the security of fingerprint recognition systems and requires design of robust and generalizable PA detectors. It is observed that the selection of PA materials used in training (known PAs) directly impacts the performance against unknown PAs, however the underlying reasons for this phenomena are unknown. In this study, we investigate the PA material characteristics and correlate them with the 3D t-SNE embeddings of PA materials and their cross-material performances. This enables us to identify a subset of PA materials, namely Silicone, 2D Paper, Play Doh, Gelatin, Latex Body Paint, and Monster Liquid Latex essential for training a robust PAD. We posit that this approach can be utilized to estimate the PAD performance against new materials by analyzing its material characteristics and t-SNE visualization of only few samples instead of collecting large datasets for each of the new material.

Next, we propose a style-transfer based wrapper, Universal Material Generator (UMG), to improve the generalization performance of any PA detector against novel PA fabrication materials that are unknown to the system during training. The proposed approach is shown to improve the average generalization performance of two state-of-the-art PA detectors, namely Fingerprint Spoof Buster (and Slim-ResCNN), from TDR of 75.24% (73.09%) to 91.78% (90.63%) @ FDR = 0.2%, respectively, when evaluated on a large-scale dataset of $5,743$ live and $4,912$ PA images fabricated using 12 materials. Our approach also improves the average cross-sensor performance from 67.60% (64.62%) to 80.63% (77.59%) for Fingerprint Spoof Buster (Slim-ResCNN) when tested on LivDet 2017 dataset, alleviating the time and resources required to generate large-scale PA datasets for every new sensor and PA material. We have also fabricated physical PA specimens using a mixture of known PA materials to explore the role of cross-material style-transfer in improving generalization performance.

Finally, we utilize the dynamics involved in the presentation of a finger, such as skin blanching, distortion, and perspiration, to learn a robust PAD. This approach uses a sequence of local patches centered at detected minutiae from ten color frames captured at 8 fps as the finger is presented on the sensor. The proposed approach improves the PA detection performance from TDR of 81.65% to 86.20% @ FDR = 0.2% in cross-material scenarios, while retaining high performance in the known material scenario.

Chapter 4

Presentation Attack Detection for OCT Fingerprint Images

In the previous chapters, we addressed the problem of presentation attack detection and its generalization using conventional fingerprint readers, *e.g.*, optical and capacitive readers, that image the 2D surface fingerprint. In this chapter, we explore the use of optical coherent tomography (OCT) fingerprint technology which provides rich depth information, including internal fingerprint (papillary junction) and sweat (eccrine) glands, in addition to imaging any fake layers (presentation attacks) placed over finger skin. Unlike 2D surface fingerprint scans, additional depth information provided by the cross-sectional OCT depth profile scans are purported to thwart fingerprint presentation attacks. We develop and evaluate a presentation attack detector (PAD) based on a deep convolutional neural network (CNN). The input data to our CNN is local patches extracted from the cross-sectional OCT depth profile scans captured using THORLabs Telesto series spectral-domain fingerprint reader. The proposed approach achieves a TDR of 99.73% @ FDR of 0.2% on a database of $3,413$ bonafide and 357 PA OCT scans, fabricated using 8 different PA materials. By employing a visualization technique, known as *CNN-Fixations*, we are able to identify the regions in the OCT scan patches that are crucial for fingerprint PAD detection.

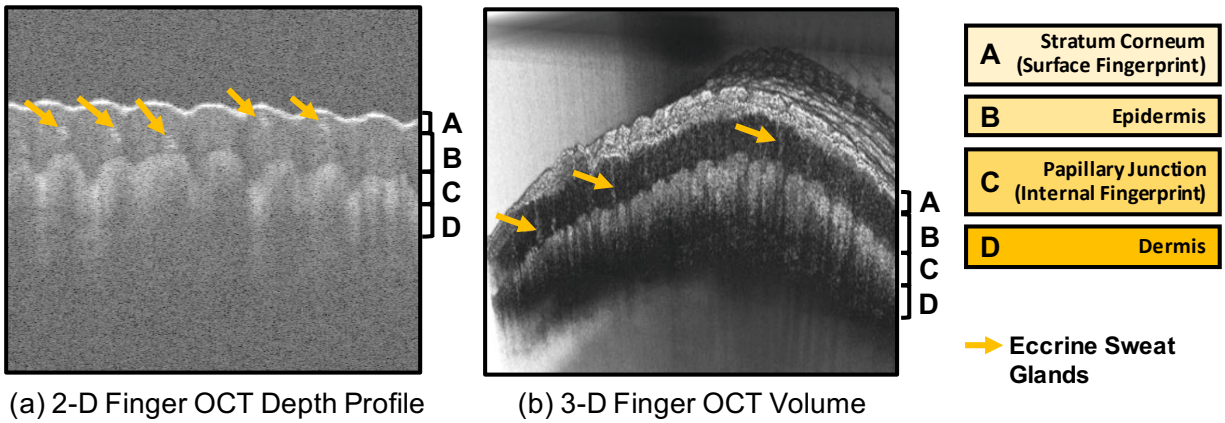


Figure 4.1 Different layers of a finger (stratum corneum, epidermis, papillary junction, and dermis) are distinctly visible in a OCT finger scan, along with helical shaped eccrine sweat glands in (a) 3-D finger OCT volume and (b) 2-D finger OCT depth profile. Note that (a) and (b) are OCT scans of different fingers. Image (a) is captured using THORLabs Telesto series (TEL1325LV2) SD-OCT scanner [154] and (b) is reproduced from [33].

4.1 Introduction

Most of the fingerprint recognition systems based on traditional readers (e.g., FTIR and capacitive technology) rely upon the friction ridge information on the finger surface (*i.e.*, stratum corneum). This makes them highly vulnerable to be fooled by presentation attacks. On the other hand, optical coherence tomography (OCT) [72] technology allows non-invasive, high-resolution, cross-sectional imaging of internal tissue microstructures by measuring their optical reflections. An optical analogue to Ultrasound [164], it utilizes low-coherence interferometry of near-infrared light ($900nm - 1325nm$) and is widely used in biomedical applications, such as ophthalmology [132], oncology [66], dermatology [162] as well as applications in art conservation [99] and fingerprint presentation attack detection [111]. In an OCT scanner, a beam of light is split into a *sample arm*, *i.e.*, a unit containing the object of interest, and a *reference arm*, *i.e.*, a unit containing a mirror to reflect back light without any alteration (see Fig. 4.2). If the reflected light from the two arms are within coherence distance, it gives rise to an interference pattern representing the depth profile at a single point, also known as *A-scan*. Laterally combining a series of A-scans along a line can provide a cross-sectional scan, also known as *B-scan* (see Figs. 4.1 (b) and 4.3). Stacking multiple

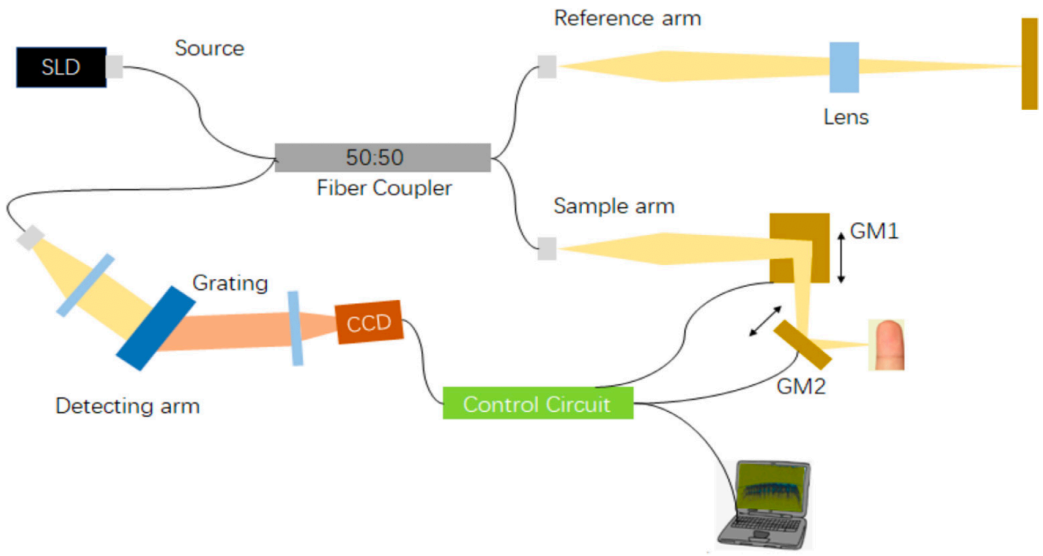


Figure 4.2 A schematic diagram of a spectral-domain optical coherent tomography (SD-OCT) scanner. The source light is emitted by a super luminescent diode (SLD) which is split into a sample arm and a reference arm. A high-resolution tomography image of the internal microstructure of the biological tissue is performed by measuring the interference signal of the sample backscattered light. Image reproduced from [100].

B-scans together can provide a 3D volumetric representation of the scanned object, or the object of our interest *i.e.*, internal structure of a finger (see Figure 4.1 (a)).

The human skin is a layered tissue with the outermost layer known as *epidermis* and the external-facing sublayer of epidermis, where the friction ridge structure exists, is known as *stratum corneum*. The layer below epidermis is known as *dermis*, and the junction between epidermis and dermis layers is known as *papillary junction*. The development of friction ridge patterns on papillary junction, which starts as early as in weeks 10-12 of gestation, results into the formation of a surface fingerprint on stratum corneum [7]. The surface friction ridge pattern, scanned by traditional (optical and capacitive) fingerprint readers, is merely an instance or a projection of the, so to say, a *master print* existing on the papillary junction. There also exist helically shaped ducts in epidermis layer connecting the eccrine (sweat) glands in dermis to the sweat pores on surface. See Figure 4.1.

Table 4.1 Existing studies on Optical Coherent Tomography (OCT) based fingerprint presentation attack detection.

Study	Approach	OCT Technology	Database	Comments
Cheng et al., 2006 [19]	Averaged B-scan slices to generate 1D depth profile; performed auto-correlation analysis; B-scan is 2.2mm in depth and 2.4mm laterally	Imalux Corp. Time-domain OCT; capture time: 3s	8 bonafide (8 fingers of one subject) and 10-20 impressions per PA, four PA materials	Manual inspection of auto-correlation response
Cheng et al., 2007 [20]	Extended [19] by combining 100 B-Scans to create 3D representation; anisotropic resolution (4762 dpi, 254 dpi)	Imalux Corp. Time-domain OCT; capture time: 300s for 100 scans	One bonafide finger, one PA	Visual analysis of 3D representation
Bosen et al., 2010 [10]	Used fingerprint COTS for matching 3D OCT scans; scanned volume: 14mm x 14mm x 3mm; discussed detection of eccrine glands for PAD	THORLabs Swept-source OCT (OCS1300SS); capture time: 20s for 3D volume	153 impressions from 51 fingers for identification experiment; one PA material.	Visual analysis for PAD; identification performance: FRR = 5% @ FAR = 0.01%
Liu et al., 2010 [102]	Mapped subsurface eccrine glands with sweat pores on finger surface; exhibited repeatable matching of fingerprints based on sweat pores; discussed absence of sweat pores for fingerprint PAD	Custom Spectral-domain OCT; capture time: 4min for 3D volume	Nine bonafide impressions from three fingers, two PA materials	Visual analysis of eccrine glands for PAD
Nasiri-Avanaki et al., 2011 [116]	Used a dynamic focus *en-Face* OCT to detect any layer placed over finger skin; discussed Doppler OCT to detect blood flow and sweat production for liveness detection	Custom *en-Face* OCT; capture time is not reported	One bonafide finger, one PA	Visual analysis of one bonafide finger and one sellotape PA
Liu et al., 2013 [101]	Auto-correlation analysis between adjacent B-Scans to determine blood flow in micro-vascular pattern	Swept-source OCT; capture time: 20s	One bonafide with and w/o inhibited blood flow	Exhibited repeatable signs of vitality
Meissner et al., 2013 [109]	Detected number of helical eccrine gland ducts to distinguish bonafide vs PA, scanned volume: 4.5mm x 4mm x 2mm	Swept-source OCT; capture time is not reported	Bonafide: $7,458$ images, cadavers: 330 images, PA: $2,970$ images	Manual PAD: 100%; automated PAD: bonafide: 93% and PA: 74% success rate
Darlow et al., 2016 [33]	Detected double bright peaks in depth profile for thin PAs and autocorrelation analysis for thick PAs; 2 different resolutions; scanned volume: 13mm x 13mm x 3mm (500dpi) and 15mm x 15mm x 3mm (867 dpi)	THORLabs Swept-source OCT (OCS1300SS); capture time: 20s for 3D volume	Bonafide: 540 scans from 15 subjects, PA: 28 scans; one PA material + sellotape	PAD accuracy: 100%
Darlow et al., 2016 [32]	Measured ridge frequency consistency of the internal fingerprint in non-overlapping blocks;	THORLabs Swept-source OCT (OCS1300SS)	Bonafide: 20 scans, PA 20 scans; one PA material	PAD accuracy: 100%
Liu et al., 2019 [100]	Analyzed order and magnitude of bright peaks in 1-D depth signals to detect PAs with different thickness; scanned volume: 15mm x 15mm x 1.8mm	Custom Spectral-domain OCT	Bonafide: 30 scans from 15 subjects, PA: 60 scans; four PA materials	Contact-based (glass platen) OCT scanner; PAD accuracy: 100%
Proposed Approach	Trained a deep CNN model using overlapping patches extracted from detected finger depth profile in B-Scans; B-scan is 1.8mm in depth and 14mm laterally	THORLabs Spectral-domain OCT (TEL1325LV2); capture time: < 1s	Bonafide: 3,413 scans from 415 subjects, PA: 357 scans, eight PA materials	Five-fold cross-validation; TDR = 99.73% @ FDR = 0.2%

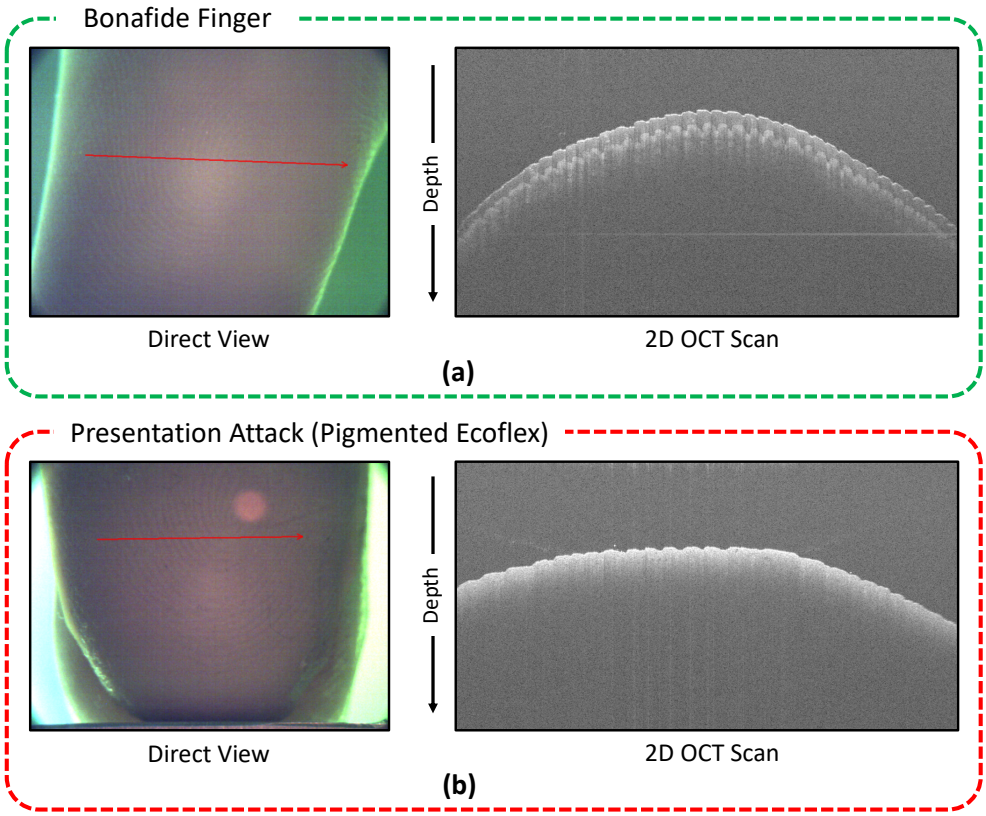


Figure 4.3 Direct view images with red arrows presenting the scanned line and the corresponding cross-sectional B-scan for a (a) bonafide and a (b) pigmented ecoflex presentation attack.

4.1.1 Related Work

Since OCT enables imaging the 3D volumetric morphology of the skin tissue, including the subsurface fingerprint and other internal structures, it has great potential in detecting fingerprint presentation attacks. Existing fingerprint PAD studies in the literature have explored various OCT technologies such as time-domain, fourier-domain, and spectral domain, and developed hand crafted features to detect blood flow, eccrine glands, and correlation between the surface and internal fingerprint.

In 2006, Cheng et al. [19] utilized a time-domain OCT scanner to capture B-scan slices which were averaged to generate 1D depth profile signals. They used auto-correlation of 1D signals to manually distinguish bonafide from PA. Stacking 100 B-scan slices allowed them to create a 3D representation of the internal finger structure for better visualization to distinguish between live

and PA [20]. In 2010, Bosen et al. [10] utilized a swept-source OCT to collect 153 impressions from 51 fingers, and a COTS matcher to evaluate the identification performance. They discussed the idea of detecting eccrine glands for PAD. Liu et al. [102] mapped subsurface eccrine glands with sweat pores on finger surface captured using a spectral-domain OCT and discussed the idea of using absence of sweat pores for PAD. In 2011, Nasiri-Avanaki et al. [116] utilized a custom dynamic focus *en-Face* OCT capable to capture any layer placed over finger skin and also discussed a method to utilize Doppler OCT for detecting blood flow and sweat production for PAD.

In 2013, Liu et al. [101] used a swept-source OCT to capture B-scans and used auto-correlation between adjacent scans to determine blood-flow in micro-vascular patterns. They utilized one bonafide finger with and without inhibited blood flow to show the significant changes in auto-correlation values. Meissner et al. [109] presented the first large-scale OCT-based PAD evaluation with $7,458$ bonafide images, 330 cadaver images, and $2,970$ PA images captured using a swept-source OCT scanner. They utilized detection of helically shaped eccrine gland ducts and achieved 100% PAD performance on manual analysis. However, the detection rates dropped to 93% and 74% for bonafide and PA, respectively, using an automated algorithm. In 2016, Darlow et al. [33] utilized swept-source OCT 1D scans and detected double bright peaks for thin PAs and analyzed auto-correlation for thick PAs. A perfect PAD accuracy was achieved with 28 PA scans and 540 bonafide scans from 15 subjects. However, they only utilized 1 PA material for thick PA and sellotape for thin PA. In [32], Darlow et al. measured ridge frequency consistency of internal fingerprints in non-overlapping blocks. They used 20 bonafide and 20 PA scans fabricated using 1 PA material and achieved 100% accuracy. Recently, in 2019, Liu et al. [100] utilized a custom spectral-domain OCT scanner and analyzed order and magnitude of bright peaks in 1D depth profile signals to detect PAs. These studies are summarized in Table 4.1.

In the proposed approach, we utilize local patches (150×150) extracted from the automatically segmented finger depth profile from input B-scan images to train a deep convolutional neural network. The main contributions of this chapter are:

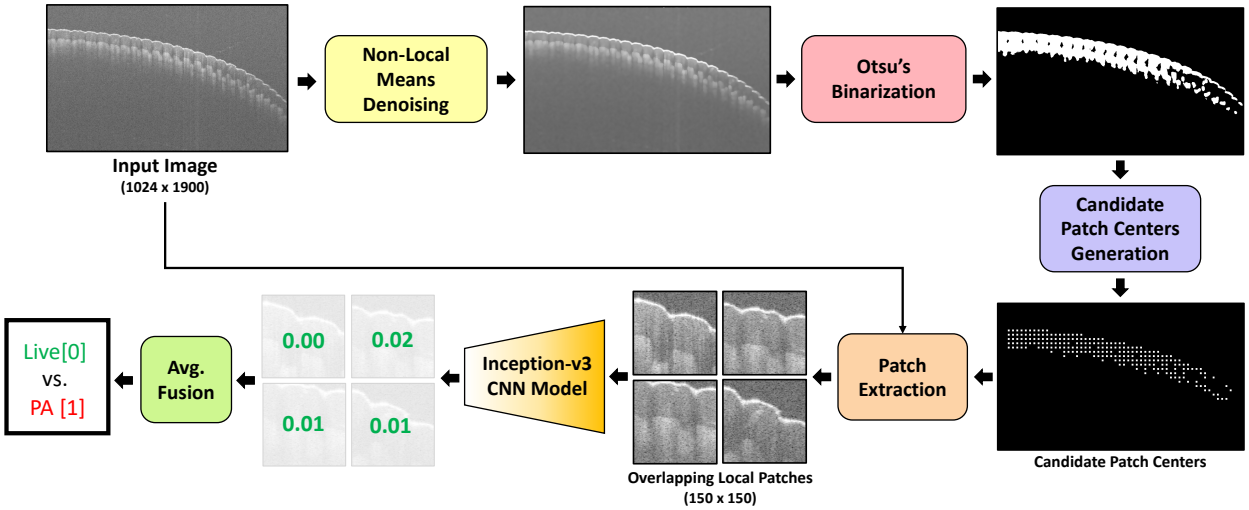


Figure 4.4 An overview of the proposed fingerprint presentation attack detection approach utilizing local patches extracted from the segmented depth profiles from OCT B-scans.

1. Proposed a deep convolutional neural network based PAD approach trained on local patches containing finger depth profile from cross-sectional B-Scans.

2. Evaluated the proposed approach on a database of 3,413 bonafide and 357 PA OCT B-scans fabricated using 8 different PA materials and achieved a TDR of 99.73% @ FDR of 0.2% for PAD.

3. Identified the regions in the OCT scan patches that are crucial for fingerprint PAD detection by employing a visualization technique, known as *CNN-Fixations*.

4.2 Proposed Approach

The proposed PAD approach includes two stages, an offline training stage and an online testing stage. The offline training stage involves (i) preprocessing the OCT images (noise removal and image enhancement), (ii) detecting region-of-interest (*i.e.*, finger depth profile), (iii) extracting local patches from the region-of-interest (ROI), and (iv) training CNN models on the extracted local patches. During the online testing stage, the final spoof detection decision is made based on the average of spoofness scores output from the CNN model for each of the extracted patches. An overview of the proposed approach is presented in Figure 4.4.

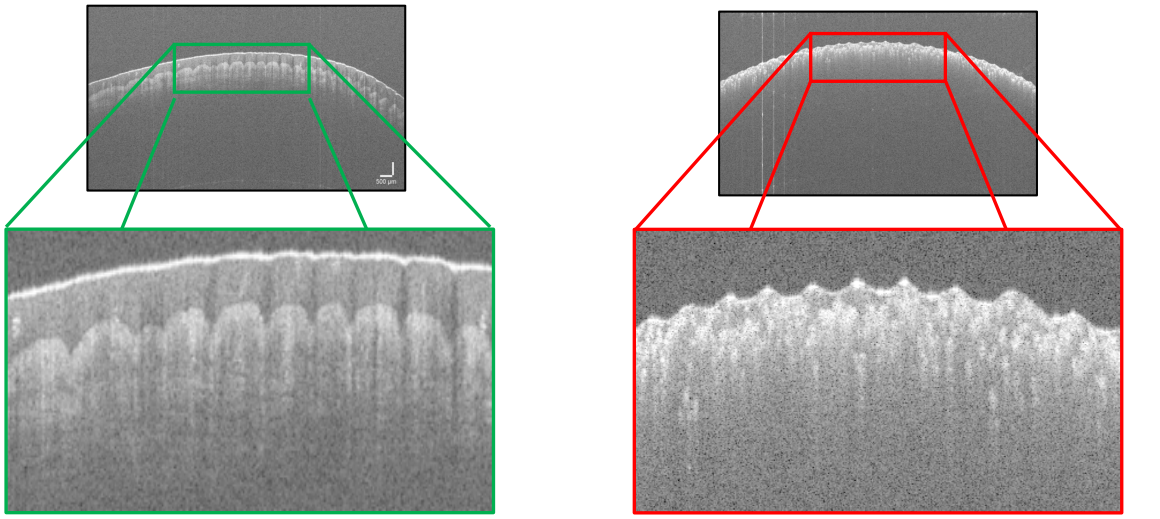


Figure 4.5 Depth profile of a bonafide finger manifests a layered tissue anatomy quite distinguishable from the depth profile of a presentation attack without any specific structure.

4.2.1 Preprocessing

Optical Coherent Tomography (OCT) 2D scans are grayscale images with height $= 1024$ pixels and width $= 1900$ pixels (see Figs. 4.5 and 4.6). These images contain gaussian noise which makes the extraction of region-of-interest (finger depth profile) by simple thresholding prone to errors. We employ Non-Local Means denoising [12] that removes noise by replacing the intensity of a pixel with an average intensity of the similar pixels that may not be present close to each other (non-local) in the image. An optimized opencv python implementation[1] of Non-Local Means denoising, *cv2.fastNlMeansDenoising()*, is used with *filterStrength* = 20, *templateWindowSize* = 7, and *searchWindowSize* = 21. After de-noising, a morphological operation of image dilation [49], with the kernel size of 5×5, is applied to enhance the image.

[1] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_photo/py_non_local_means/py_non_local_means.html

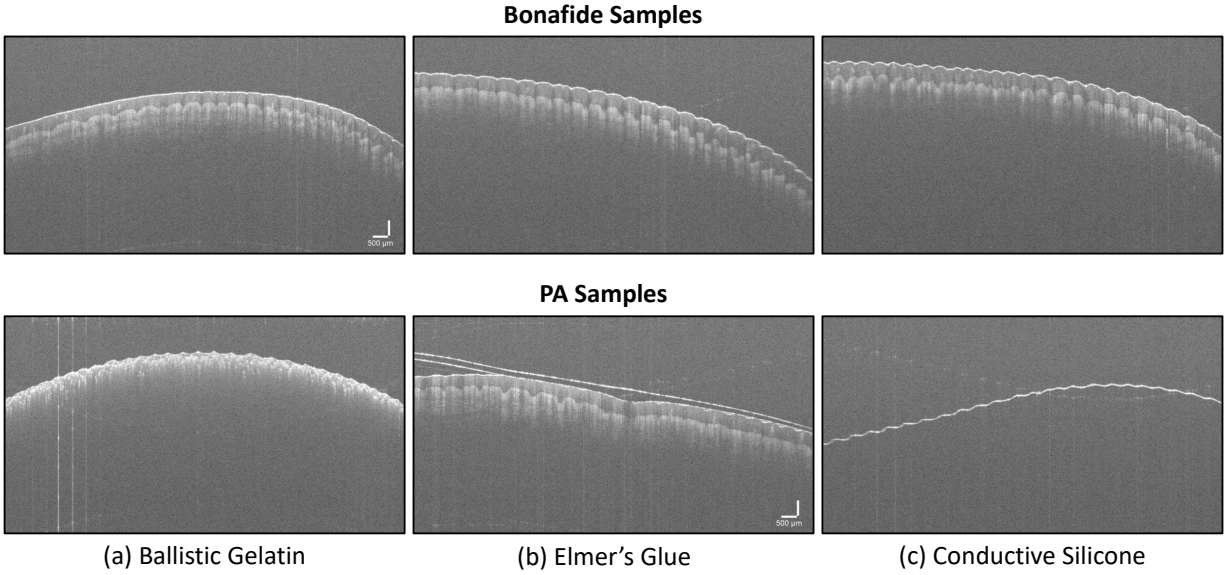


Figure 4.6 Examples of bonafide and presentation attack samples from the OCT fingerprint database utilized in this study.

4.2.2 Otsu's Binarization

The characteristic differences between a bonafide and a presentation attack OCT image are primarily discernible in the finger depth profile region as shown in Figure 4.5. The pixel intensity histograms for the grayscale 2D OCT images are bimodal, with the first peak (high intensity values) referring to the finger depth profile region, while the second peak (low intensity values) refers to the background region. In order to segment out the finger depth profile, we apply Otsu's thresholding [125] which finds an adaptive threshold, in the middle of the two peaks, to successfully binarize the input OCT images as shown in Figure 4.4.

4.2.3 Local Patch Extraction

The binarized image generated after Otsu's binarization is raster scanned, with a stride of 30 pixels (in both x and y-axis), to identify the possible candidates for patch extraction. At each scanned pixel, a window of size 9×9 is evaluated and if more than 25% of the pixels (20 out of 81 pixels) in the window have non-zero values, the pixel is marked as a candidate for extracting a local patch.

Table 4.2 Summary of the Optical Coherent Tomography (OCT) database collected at GCT-II as part of IARPA ODIN Program [123].

Fingerprint Presentation Attack Material	**#Images**
Ballistic Gelatin	34
Clear Ecoflex	7
Tan Ecoflex	49
Yellow Pigmented Silicone	57
Flesh Pigmented Ecoflex	36
Nusil R-2631 Conductive Silicone	128
Flesh Pigmented PDMS	42
Elmer's Glue	1
Bandaid	3
Total PAs	**357**
Total Bonafide	**3,413**

This rule is applied to guarantee sufficient depth information in the extracted patches. After the patch candidates are selected, a maximum of 60 local patches of size 150×150 are extracted from the original image around the patch candidates. If there are more than 60 candidates, the topmost candidates from each column (*i.e.*, the points closest to the surface fingerprint) are selected first, before moving to the next row. With the image width of 1900 pixels and a stride of 30 pixels, a maximum of 60 patches are sufficient to provide at least one pass of *stratum corneum*. The patches are extracted such that the candidate is located at $(50, 75)$ in the 150×150 patch. This ensures that the extracted patches cover stratum corneum, epidermis, and papillary junction regions as shown in Figure 4.4.

4.2.4 Convolution Neural Networks

With the success of AlexNet [93] in ILSVRC-2012 [140], different deep CNN architectures have been proposed in literature, such as VGG, GoogleNet (Inception), Inception v2-v4, MobileNet, and ResNet. In this study, we utilize the Inception-v3 [150] architecture which has exhibited

state-of-the-art performance in patch-based fingerprint presentation attack detection [23, 24]. Our experimental results show that training the models from scratch, using local patches, performs better than fine-tuning a pre-trained network on image patches from other domains (e.g. FTIR fingerprint images).

We utilized the TF-Slim library[2] implementation of the Inception-v3 architecture. The last layer of the architectures, a 1000-unit softmax layer (originally designed to classify a query image into one of the the $1,000$ classes of ImageNet dataset) was replaced with a 2-unit softmax layer for the two-class problem, *i.e.*, Bonafide vs. PA. The output from the softmax layer is in the range $[0, 1]$, defined as *Spoofness Score*. The larger the spoofness score, the higher the likelihood that the input patch belongs to the PA class. For an input test image, the spoofness scores corresponding to each of the local patches, extracted from the input image, are averaged to give a *Global Spoofness Score*. The optimizer used to train the network was RMSProp, with a batch size of 32, and an adaptive learning rate with exponential decay, starting at 0.01 and ending at 0.0001. Data augmentation techniques, such as random cropping, brightness adjustment, horizontal and vertical flipping, are employed to ensure the trained model is robust to the possible variations in fingerprint images. The proposed approach is presented in Algorithm 2.

4.3 Experimental Results

4.3.1 OCT Presentation Attack Database

A database of $3,413$ bonafide and 357 presentation attack (PA) 2D OCT scans is utilized in this study. These scans are captured using THORLabs Telesto series (TEL1325LV2) Spectral-domain OCT scanner [154] (see Figure 4.7). Table 4.2 lists the eight PA materials and the corresponding number of scans for each material type. Figure 4.6 presents few samples of bonafide and PA scans

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

Algorithm 2 Presentation Attack Detection for OCT Fingerprint Images

procedure
input
    I: 2D OCT Fingerprint Image
output
    S_I: Predicted Spoofness Score for I
functions and parameters
    $f(.)$: OpenCV non-local means denoising function *cv2.fastNlMeansDenoising()*
    θ_f: $filterStrength = 20$, $templateWindowSize = 7$, and $searchWindowSize = 21$
    $g(.)$: image dilation function
    θ_g: $kernelSize = 5$
    $h(.)$: Otsu's Binarization Function
    $p(.)$: Raster-scan local patch extractor with maximum number of patches = 60
    θ_p: $h = w = 150$, $Stride_x = 30$, $Stride_y = 30$, $PatchCenter = (50, 75)$
    $c(.)$: Inception-v3 CNN Model trained on Bonafide and PA OCT patch images, returns spoofness scores for input patches
begin:
    Preprocessing: $I_p = g(f(I, \theta_f), \theta_g)$
    Binarized Image: $I_b = h(I_p)$
    Local Patch Extraction: $\phi = p(I, I_b, \theta_p)$
    CNN Evaluation of Local Patches: $S_\phi = c(\phi)$
    Spoofness Score: $S_I = average(S_\phi)$
end

from this database. This dataset is collected at John Hopkins University Applied Physics Lab[3] as part of a large-scale evaluation under IARPA ODIN Project [123] on presentation attack detection.

4.3.2 Results

The proposed approach is evaluated using five-fold cross-validation. Table 4.3 presents the training and testing set details for each fold[4], along with the achieved PA True Detection Rate (%) @ False Detection Rate $= 0.2\%$. The selection of this metric is based on the requirements of IARPA ODIN program [123] and represents the percentage of PAs able to breach the biometric system security when the reject rate of legitimate users $\leq 0.2\%$. Note that the proposed approach achieves an avg. TDR = 99.73% (s.d. = 0.55) @ FDR = 0.2% for the five folds. Figure 4.8 presents the ROC

[3]https://www.jhuapl.edu/

[4]Note that all PA types are uniformly distributed among the five folds without repetition, therefore Elmer's Glue and Bandaid which have less than five samples are missing from some folds.

Figure 4.7 Setup of a THORLabs Telesto series Spectral-domain OCT scanner (TEL1325LV2). Image taken from [154].

Table 4.3 Summary of the five-fold cross-validation and the performance achieved using Inception-v3 model.

Fold	**# Images (Bonafide / PA)**		**TDR (%) @ FDR = 0.2%**
	Training	**Testing**	
I	(2,730 / 281)	(683 / 76)	100.00
II	(2,730 / 283)	(683 / 74)	98.63
III	(2730 / 288)	(683 / 71)	100.00
IV	(2731 / 289)	(682 / 70)	100.00
V	(2731 / 288)	(682 / 71)	100.00
Average			**99.73** (s.d. = 0.55)

curves for each of the five folds. In fold-II, only one bonafide scan was misclassified as PA due to incorrect segmentation.

4.3.3 Visualizing CNN Learnings

CNNs have revolutionized computer vision and machine learning research achieving unprecedented performance in many tasks. But these are usually treated as "black boxes" shedding little light on their internal workings and without answering how they achieve high performance. One way to gain insights into what CNNs learn is through visual exploration, *i.e.*, to identify the

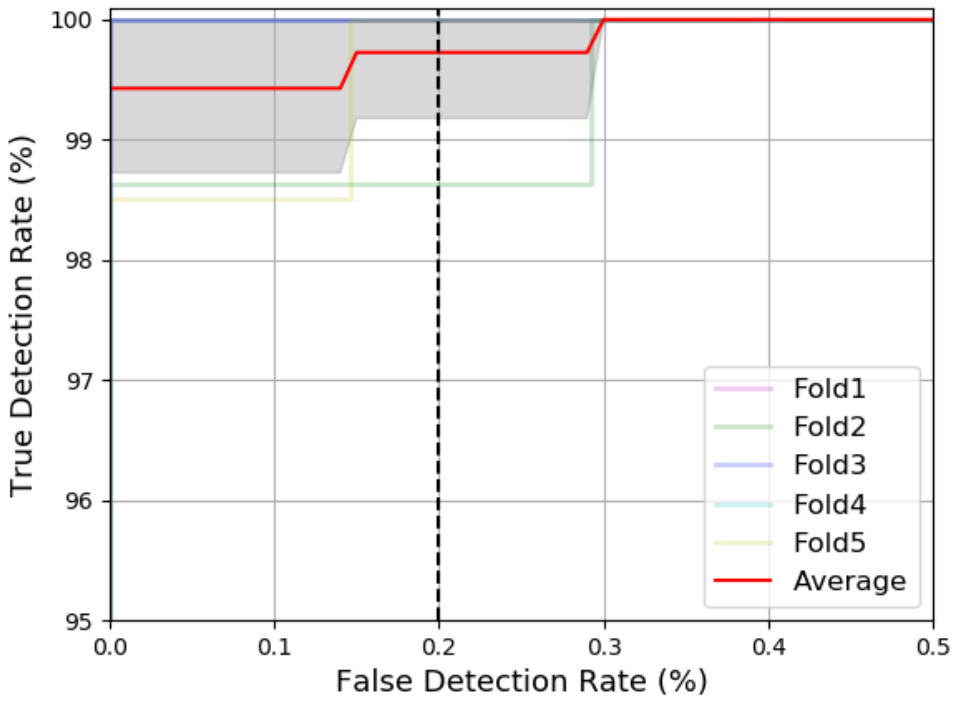


Figure 4.8 ROC curves for the five-fold cross-validation experiments. The red curve represents the average performance with grayed region reflecting the confidence interval of one standard deviation.

image regions that are responsible for the final predictions. Towards this goal, visualization techniques [112, 144, 146] have been proposed to supplement the class labels predicted by CNN, in our case bonafide or PA, with the discriminated image regions (or saliency maps) exhibiting class-specific patterns learned by CNN architectures. The visualization technique proposed in [112] exploits the learned feature dependencies between consecutive layers of a CNN to identify the discriminative pixels, called *CNN-Fixations*, in the input image that are responsible for the predicted label. We utilize this visualization technique to understand the representation learning of our CNN models and identify the crucial regions in OCT images responsible for final predictions. Figs. 4.9 presents CNN-Fixations and the corresponding density heatmaps for two bonafide and two PA image patches that are correctly classified. We observe that there is a high density of fixations along stratum corneum and at papillary junction, suggesting that these are definitely crucial regions in distinguishing bonafide vs PA OCT patches. Note that the only misclassified sample in Fold-II was

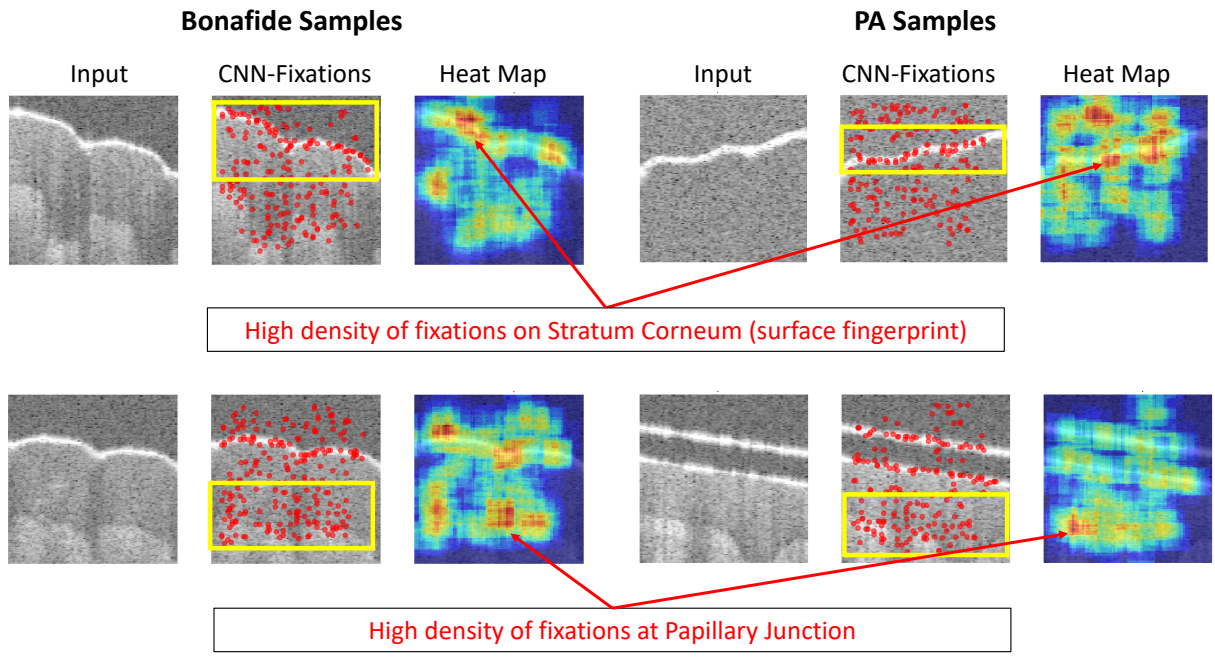


Figure 4.9 Patches (150×150) from bonafide and PA OCT B-scans input to the model are presented. The detected CNN-Fixations and a heat map presenting the density of CNN-Fixations are also shown. A high density of fixations are observed along the stratum corneum (surface fingerprint) and at papillary junction in both bonafide and PA patches. (Best viewed in color)

due to incorrect segmentation, otherwise it would be useful to observe the CNN-Fixations that led to an incorrect prediction.

4.4 Summary

The penetrative power of optical coherent tomography (OCT) to image the internal tissue structure of human skin in a non-invasive manner presents a great potential to investigate robustness against fingerprint presentation attacks. We propose and demonstrate a learning-based approach to differentiate between bonafide (live) and eight different types of presentation attacks (spoofs). The proposed approach utilizes local patches automatically extracted from the finger depth profile in 2D OCT B-scans to train an Inception-v3 network model. Our experimental results achieve a TDR of 99.73% @ FDR of 0.2% on a database of $3,413$ bonafide and 357 PA scans. The crucial regions in the input images for PAD learned by the CNN models, namely *stratum corneum* and

papillary junction, are identified using a visualization technique. In future, we will evaluate the generalization ability of the proposed approach against novel materials that are not seen by the model during training.

Chapter 5

Summary

In this thesis, we address the challenges of presentation attack detection by developing an *accurate*, *efficient*, *interpretable*, and *generalizable* solution to detect fake/gummy fingers (spoofs) and altered fingerprints. The proposed solution achieves state-of-the-art accuracy on publicly available liveness detection (LivDet) databases, large-scale government (IARPA ODIN program) evaluation databases, two new in-house self-collected databases, and an operational altered fingerprint database from a law enforcement agency. Fingerprints used in these datasets are captured using both traditional fingerprint readers, *e.g.*, CrossMatch Guardian 200, Lumidigm V302, SilkID Fast Frame Rate, etc., as well as novel fingerprint readers based on optical coherent tomography (OCT). The proposed solution is optimized, in terms of both memory and computational resources, for real-time inference and is ported as an efficient Android app that can make a PAD decision in under 100ms on a commodity smartphone (Samsung Galaxy S8). Furthermore, we investigate the optical and physical characteristics of different spoof materials to understand and interpret the cross-material (generalization) performance achieved by the proposed approach. We also improve the PAD generalization performance by proposing two difference approaches: (i) a style transfer-based wrapper to generate spoof images of unknown styles and (ii) a temporal analysis of a sequence of fingerprint image frames.

5.1 Contributions

The main contributions of this thesis are summarized below:

1. An accurate deep learning-based fingerprint presentation attack detector (PAD), called *Fingerprint Spoof Buster*, utilizing local patches centered and aligned along fingerprint minutiae. The proposed approach utilizing only grayscale fingerprint images can be integrated as a *software-only solution*, without incurring any additional hardware cost, to a wide range of already deployed fingerprint matching systems. Our algorithm can be generalized to images captured by any sensor with minimal retraining.

2. A graphical user interface for the Fingerprint Spoof Buster which highlights the local regions of the fingerprint image as bonafide (live) or PA (spoof) for visual inspection. This is more informative than a single spoof score output by the traditional approaches. We utilize visualization techniques to interpret the features learned by CNN models in order to understand the strengths and limitations of the proposed approach. In the same spirit, we also propose a method for detection and localization of fingerprint alterations utilizing whole images and minutia-centered patches to train CNN models, achieving state-of-the-art accuracy.

3. We tackle the high memory and computational requirements of Fingerprint Spoof Buster by (i) minutiae clustering, followed by weighted fusion to reduce the required number of local patch inferences, and (ii) optimizing the network architecture and quantization of model weight parameters to perform byte computations instead of floating point arithmetic. The proposed optimizations result in an approximately 80% reduction in computation and memory requirements. This has enabled us to develop a light-weight version of the PAD, called *Fingerprint Spoof Buster Lite*, as an Android application that can run on a commodity smartphone (Samsung Galaxy S8) without a significant drop in PAD performance (from TDR = 95.7% to 95.3% @ FDR = 0.2%) capable of detecting spoofs in under 100ms.

4. An interpretation of cross-material (generalization) performance of the proposed PAD by (i) evaluating Fingerprint Spoof Buster against unknown PAs by adopting a leave-one-out protocol where one material is left out from training set and set aside for testing, (ii) utilizing 3D t-SNE visualizations of the bonafide and PA samples in the deep feature space, and (iii) investigating the PA material characteristics (two optical and two physical properties) and correlating them with their cross-material performances to identify a representative set of PA materials that should be included during training to ensure a high generalization performance.

5. A style transfer-based wrapper, *Universal Material Generator* (UMG), to improve the generalization performance of any PA detector against novel PA fabrication materials that are unknown to the system during training. The proposed wrapper is shown to improve the average generalization performance of Fingerprint Spoof Buster from TDR of 75.24% to 91.78% @ FDR = 0.2% when evaluated on a large-scale dataset of $5,743$ live and $4,912$ PA images fabricated using 12 materials. It is also shown to improve the average cross-sensor performance from 67.60% to 80.63% when tested on LivDet 2017 dataset, alleviating the time and resources required to generate large-scale PA datasets for new sensors.

6. A dynamic PAD solution utilizing a sequence of local patches centered at detected minutiae from ten color frames captured in quick succession (8 fps) as the finger is presented on the sensor. We posit that the dynamics involved in the presentation of a finger, such as skin blanching, distortion, and perspiration, provide discriminating cues to distinguish live from spoofs. The proposed approach improves the spoof detection performance from TDR of 99.11% to 99.25% @ FDR = 0.2% in known-material scenarios, and from TDR of 81.65% to 86.20% @ FDR = 0.2% in cross-material scenarios.

7. A PAD solution utilizing the ridge-valley depth-information of finger skin, including internal fingerprint (papillary junction) and sweat (eccrine) glands, sensed by the optical coherent tomography (OCT) fingerprint technology. Our proposed solution achieves a TDR of 99.73%

@ FDR of 0.2% on a database of $3,413$ bonafide and 357 PA OCT scans captured using THORLabs Telesto series spectral-domain fingerprint reader. We also identify the regions in the OCT scan patches that are crucial for fingerprint PAD detection.

5.2 Suggestions for Future Work

The following are some of the possible future directions within the scope of fingerprint presentation attack detection:

- **PAD Generalization**: Explore adversarial representation learning (ARL) based approach [139] to learn *material* and *sensor* agnostic feature representations for generalized fingerprint PAD.

- **Multi-Task Learning**: In addition to detecting bonafide vs. PA, a PAD could be trained to predict the PA material type as an open-set problem. With one of the classes as "unknown" material, the system could be trained in a continuous (online) manner when the network is not able to predict the material type with high confidence.

- **Dynamic PAD Approaches**: Learning a *"mixture of PAD experts"* where each expert module specializes in some sensor and/or some PA materials. The selection of the best module can be learned as an auxiliary task and this decision can be made dynamically at test time. s

- **Altered Fingerprints**: Explore GAN-based generative models and 3D printing of altered fingerprint targets to increase the availability of altered fingerprint databases in the literature for conducting a large-scale study.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Meir Agassy, Boaz Castro, Arye Lerner, Gal Rotem, Liran Galili, and Nathan Altman. Liveness and Spoof Detection for Ultrasonic Fingerprint Sensors, April 16 2019. US Patent 10,262,188.

[2] Athos Antonelli, Raffaele Cappelli, Dario Maio, and Davide Maltoni. Fake Finger Detection by Skin Distortion Analysis. *IEEE Transactions on Information Forensics and Security*, 1(3):360–373, 2006.

[3] Apple. Apple Pay: Payment authorization using Touch ID. https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf, May 2019.

[4] 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, 2016.

[5] 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*, 12(9):2067–2077, 2017.

[6] David R. Ashbaugh. *Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology*. CRC press, 1999.

[7] William J. 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 *Proc. International Conference on Biometrics (ICB)*, pages 265–272. Springer, 2006.

[9] Mauro Barni et al. A Privacy-compliant Fingerprint Recognition System based on Homomorphic Encryption and Fingercode Templates. In *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–7, 2010.

[10] Anke Bossen, Roland Lehmann, and Christoph Meier. Internal Fingerprint Identification with Optical Coherence Tomography. *IEEE Photonics Technology Letters*, 22(7):507–509, 2010.

[11] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[12] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-local Means Denoising. *Image Processing On Line*, 1:208–212, 2011.

[13] Kai Cao and Anil K. Jain. Learning Fingerprint Reconstruction: From Minutiae to Image. *IEEE Transactions on Information Forensics and Security*, 10(1):104–117, 2014.

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

[15] Kai Cao and Anil K. Jain. Automated Latent Fingerprint Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):788–800, 2018.

[16] Kai Cao, Eryun Liu, Liaojun Pang, Jimin Liang, and Jie Tian. Fingerprint Matching by Incorporating Minutiae Discriminability. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–6, 2011.

[17] 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.

[18] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet Image Montage. *ACM Transactions on Graphics (TOG)*, 28(5):124, 2009.

[19] Yezeng Cheng and Kirill V. Larin. Artificial Fingerprint Recognition by using Optical Coherence Tomography with Autocorrelation Analysis. *Applied Optics*, 45(36):9238–9245, 2006.

[20] Yezeng Cheng and Kirill V. Larin. In Vivo Two-and Three-dimensional Imaging of Artificial and Real Fingerprints With Optical Coherence Tomography. *IEEE Photonics Technology Letters*, 19(20):1634–1636, 2007.

[21] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv preprint arXiv:1610.02357*, 2016.

[22] Tarang Chugh, Sunpreet S. Arora, Anil K. Jain, and Nicholas G. Paulter. Benchmarking Fingerprint Minutiae Extractors. In *IEEE International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–8, 2017.

[23] Tarang Chugh, Kai Cao, and Anil K. Jain. Fingerprint Spoof Detection using Minutiae-based Local Patches. In *Proc. IEEE International Joint Conference on Biometrics (IJCB)*, pages 581–589, 2017.

[24] 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.

[25] Tarang Chugh, Kai Cao, Jiayu Zhou, Elham Tabassi, and Anil K. Jain. Latent Fingerprint Value Prediction: Crowd-based Learning. *IEEE Transactions on Information Forensics and Security*, 13(1):20–34, 2017.

[26] Tarang Chugh and Anil K. Jain. Fingerprint Presentation Attack Detection: Generalization and Efficiency. In *IEEE International Conference on Biometrics (ICB)*, pages 1–8, 2019.

[27] Tarang Chugh and Anil K. Jain. Fingerprint Spoof Detection: Temporal Analysis of Image Sequence. *arXiv preprint arXiv:1912.08240*, 2019.

[28] Tarang Chugh and Anil K. Jain. Fingerprint Spoof Generalization. *arXiv preprint arXiv:1912.02710*, 2019.

[29] Tarang Chugh and Anil K. Jain. OCT Fingerprints: Resilience to Presentation Attacks. *arXiv preprint arXiv:1908.00102*, 2019.

[30] European Commision. Trusted Biometrics under Spoofing Attacks (TABULA RASA). http://www.tabularasa-euproject.org/, 2013.

[31] Harold Cummins. Attempts to Alter and Obliterate Finger-Prints. *Journal of Criminal Law and Criminology*, 25(12), 1935.

[32] Luke N. Darlow, Ann Singh, Moolla, et al. Damage Invariant and High Security Acquisition of the Internal Fingerprint using Optical Coherence Tomography. In *World Congress on Internet Security*, 2016.

[33] Luke N. Darlow, Leandra Webb, and Natasha Botha. Automated Spoof-detection for Fingerprints using Optical Coherence Tomography. *Applied Optics*, 55(13):3387–3396, 2016.

[34] Dept. of Homeland Security. Office of Biometric Identity Management Identification Services. https://www.dhs.gov/obim-biometric-identification-services, 2016.

[35] 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)*, pages 1–6, 2016.

[36] FBI Criminal Justice Information Services Division. Altered Fingerprints: A Challenge to Law Enforcement Identification Efforts. www.crime-scene-investigator.net/altered-fingerprints.html, 2015.

[37] Kostadin D. Djordjev, Leonard E. Fennell, Nicholas I. Buchan, David W. Burns, Samir K. Gupta, and Sanghoon Bae. Display with Peripherally Configured Ultrasonic Biometric Sensor. US Patent 9,323,393, 2016.

[38] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A Learned Representation for Artistic Style. *arXiv preprint arXiv:1610.07629*, 2016.

[39] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. CAN: Creative Adversarial Networks, Generating "art" by Learning about Styles and Deviating from Style Norms. *arXiv preprint arXiv:1706.07068*, 2017.

[40] John Ellingsgaard and Christoph Busch. Altered Fingerprint Detection. *Handbook of Biometrics for Forensic Science, Springer*, pages 85–123, 2017.

[41] John Ellingsgaard, Ctirad Sousedik, and Christoph Busch. Detecting Fingerprint Alterations by Orientation Field and Minutiae Orientation Analysis. In *2nd International Workshop on Biometrics and Forensics*, pages 1–6, 2014.

[42] 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.

[43] Joshua J. Engelsma, Kai Cao, and Anil K. Jain. RaspiReader: Open Source Fingerprint Reader. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2511–2524, 2018.

[44] Joshua J. Engelsma, Kai Cao, and Anil K. Jain. Learning a Fixed-Length Fingerprint Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (early access)*, 2019.

[45] Joshua J. Engelsma, Debayan Deb, Anil K. Jain, Anjoo Bhatnagar, and Prem S. Sudhish. Infant-Prints: Fingerprints for Reducing Infant Mortality. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 67–74, 2019.

[46] Joshua J. Engelsma and Anil K. Jain. Generalizing Fingerprint Spoof Detector: Learning a One-Class Classifier. *IEEE International Conference on Biometrics (ICB)*, 2019.

[47] Henry Faulds. On the Skin-furrows of the Hand. *Nature*, 22(574):605, 1880.

[48] Jianjiang Feng, Anil K. Jain, and Arun Ross. Detecting Altered Fingerprints. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 1622–1625, 2010.

[49] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[50] Rohit Gajawada, Additya Popli, Tarang Chugh, Anoop Namboodiri, and Anil K. Jain. Universal Material Translator: Towards Spoof Fingerprint Generalization. In *IEEE International Conference on Biometrics (ICB)*, 2019.

[51] Francis Galton. Personal Identification and Description. *Journal of Anthropological Institute of Great Britain and Ireland*, pages 177–191, 1889.

[52] Francis Galton. *Finger Prints*. Macmillan and Company, 1892.

[53] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. *arXiv preprint arXiv:1508.06576*, 2015.

[54] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer using Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[55] Luca Ghiani, Abdenour Hadid, Gian Luca Marcialis, and Fabio Roli. Fingerprint Liveness Detection using Binarized Statistical Image Features. In *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6, 2013.

[56] Luca Ghiani, Gian Luca Marcialis, and Fabio Roli. Fingerprint Liveness Detection by Local Phase Quantization. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 537–540, 2012.

[57] Luca Ghiani, David Yambay, Valerio Mura, Gian Luca Marcialis, Fabio Roli, and Stephanie Schuckers. Review of the Fingerprint Liveness Detection (LivDet) Competition Series: 2009 to 2015. *Image and Vision Computing*, 58:110–128, 2017.

[58] Luca Ghiani, David Yambay, Valerio Mura, Simona Tocco, Gian Luca Marcialis, Fabio Roli, and Stephanie Schuckcrs. LivDet 2013 Fingerprint Liveness Detection Competition 2013. In *Proc. IAPR International Conference on Biometrics (ICB)*, pages 1–6, 2013.

[59] Lázaro J González-Soler, Marta Gomez-Barrero, Leonardo Chang, Airel Pérez-Suárez, and Christoph Busch. Fingerprint Presentation Attack Detection Based on Local Features Encoding for Unknown Attacks. *arXiv preprint arXiv:1908.10163*, 2019.

[60] Diego Gragnaniello, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Fingerprint Liveness Detection based on Weber Local Image Descriptor. In *Proc. IEEE Workshop on Biometric Meas. Syst. Secur. Med. Appl. (BIOMS)*, pages 46–50, 2013.

[61] Diego Gragnaniello, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Local Contrast Phase Descriptor for Fingerprint Liveness Detection. *Pattern Recognition*, 48(4):1050–1058, 2015.

[62] Peter W. Greenwood and Joan Petersilia. *The Criminal Investigation Process Volume I: Summary And Policy Implications*. Rand Corporation, 1975.

[63] Mark Hawthorne. *Fingerprints: Analysis and Understanding*. CRC Press, 2017.

[64] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[65] Yi He and Bo Pi. Under-screen Optical Sensor Module for On-screen Fingerprint Sensing. US Patent App. 15/421,249, 2017.

[66] Michael R. Hee, Carmen A. Puliafito, Carlton Wong, Jay S. Duker, Elias Reichel, Bryan Rutledge, Joel S. Schuman, Eric A. Swanson, and James G. Fujimoto. Quantitative Assessment of Macular Edema with Optical Coherence Tomography. *Archives of Ophthalmology*, 113(8):1019–1029, 1995.

[67] Edward R. Henry. *Classification and Uses of Finger Prints*. George Routledge and Sons, 1900.

[68] William James Herschel. Finger-Prints. *Nature*, 51(1308):77, 1894.

[69] William James Herschel. *The Origin of Finger-printing*. Oxford University Press, 1916.

[70] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation, MIT Press*, 9(8):1735–1780, 1997.

[71] 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.

[72] David Huang, Eric A. Swanson, Charles P. Lin, Joel S. Schuman, William G. Stinson, Warren Chang, Michael R. Hee, Thomas Flotte, Kenton Gregory, Carmen A. Puliafito, et al. Optical Coherence Tomography. *Science*, 254(5035):1178–1181, 1991.

[73] Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017.

[74] 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.

[75] International Standards Organization. Information Technology – Biometric Sample Quality – Part 4: Finger Image Data. https://www.iso.org/standard/62791.html, 2017.

[76] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[77] Anil K Jain. Fingerprints: Proving Ground for Pattern Recognition. In *IEEE International Conference on Pattern Recognition (ICPR)*, 2006.

[78] 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, 2016.

[79] 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.

[80] Anil K Jain and Richard C Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., 1988.

[81] Anil K. Jain and Kalle Karu. Learning Texture Discrimination Masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):195–205, 1996s.

[82] Anil K Jain, Karthik Nandakumar, and Abhishek Nagar. Biometric Template Security. *EURASIP Journal on Advances in Signal Processing*, page 113, 2008.

[83] Anil K Jain, Karthik Nandakumar, and Arun Ross. 50 Years of Biometric Research: Accomplishments, Challenges, and Opportunities. *Pattern Recognition Letters*, 79:80–105, 2016.

[84] Anil K Jain, Salil Prabhakar, and Sharath Pankanti. On the Similarity of Identical Twin Fingerprints. *Pattern Recognition*, 35(11):2653–2663, 2002.

[85] Anil K. Jain, Arun Ross, and Salil Prabhakar. Fingerprint Matching using Minutiae and Texture Features. In *Proc. IEEE Internationåal Conference on Image Processing (ICIP)*, volume 3, pages 282–285, 2001.

[86] Anil K. Jain, Arun A. Ross, and Karthik Nandakumar. *Introduction to Biometrics*. Springer Science & Business Media, 2011.

[87] Han-Ul Jang, Hak-Yeol Choi, Dongkyu Kim, Jeongho Son, and Heung-Kyu Lee. Fingerprint Spoof Detection using Contrast Enhancement and Convolutional Neural Networks. In *International Conference on Information Science and Applications*, pages 331–338. Springer, 2017.

[88] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-time Style Transfer and Super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[89] Dame Kathleen Mary Kenyon. *Archaeology in the Holy Land*. E. Benn, 1960.

[90] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[91] Jascha Kolberg, Marta Gomez-Barrero, and Christoph Busch. Multi-algorithm benchmark for fingerprint presentation attack detection with laser speckle contrast imaging. In *IEEE International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5, 2019.

[92] Peter Komarinski. *Automated Fingerprint Identification Systems (AFIS)*. Elsevier, 2005.

[93] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proc. Conference on Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[94] Philip Dean Lapsley, Jonathan Alexander Lee, David Ferrin Pare Jr, and Ned Hoffman. Anti-fraud Biometric Scanner that Accurately Detects Blood Flow, 1998. US Patent 5,737,439, 1998.

[95] Chuan Li and Michael Wand. Precomputed Real-time Texture Synthesis with Markovian Generative Adversarial Networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[96] Stan Z. Li and Anil K. Jain, editors. *Encyclopedia of Biometrics*. Springer, 2015.

[97] Xin Li, Bahadir Gunturk, and Lei Zhang. Image demosaicing: A systematic survey. In *Visual Communications and Image Processing*, volume 6822. International Society for Optics and Photonics, 2008.

[98] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying Neural Style Transfer. *arXiv preprint arXiv:1701.01036*, 2017.

[99] Haida Liang, Marta Gomez Cid, Radu G Cucu, GM Dobre, A Gh Podoleanu, Justin Pedro, and David Saunders. En-face Optical Coherence Tomography-a Novel Application of Non-invasive Imaging to Art Conservation. *Optics Express*, 13(16):6133–6144, 2005.

[100] Feng Liu, Guojie Liu, and Xingzheng Wang. High-accurate and Robust Fingerprint Anti-spoofing System using Optical Coherence Tomography. *Expert Systems with Applications*, 130:31–44, 2019.

[101] Gangjun Liu and Zhongping Chen. Capturing the Vital Vascular Fingerprint with Optical Coherence Tomography. *Applied Optics*, 52(22):5473–5477, 2013.

[102] Mengyang Liu and Takashi Buma. Biometric Mapping of Fingertip Eccrine Glands with Optical Coherence Tomography. *IEEE Photonics Technology Letters*, 22(22):1677–1679, 2010.

[103] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[104] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. *Handbook of Fingerprint Recognition*. Springer Science & Business Media, second edition, 2009.

[105] Emanuela Marasco and Arun Ross. A Survey on Antispoofing Schemes for Fingerprint Recognition Systems. *ACM Computing Surveys*, 47(2):28, 2015.

[106] Emanuela Marasco and Carlo Sansone. Combining Perspiration-and Morphology-based Static Features for Fingerprint Liveness Detection. *Pattern Recognition Letters*, 33(9):1148–1156, 2012.

[107] Sébastien Marcel, Mark S. Nixon, Julian Fierrez, and Nicholas Evans, editors. *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection*. Springer, second edition, 2019.

[108] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino. Impact of Artificial Gummy Fingers on Fingerprint Systems. In *Proc. SPIE*, volume 4677, pages 275–289, 2012.

[109] Sven Meissner, Ralph Breithaupt, and Edmund Koch. Defense of Fake Fingerprint Attacks using a Swept Source Laser Optical Coherence Tomography Setup. In *Frontiers in Ultrafast Optics: Biomedical, Scientific, and Industrial Applications XIII*, volume 8611. SPIE, 2013.

[110] David Menotti, Giovani Chiachia, Allan Pinto, William Robson Schwartz, Helio Pedrini, Alexandre Xavier Falcao, and Anderson Rocha. Deep Representations for Iris, Face, and Fingerprint Spoofing Detection. *IEEE Transactions on Information Forensics and Security*, 10(4):864–879, 2015.

[111] Yaseen Moolla, Luke Darlow, Ameeth Sharma, Ann Singh, and Johan Van Der Merwe. Optical Coherence Tomography for Fingerprint Presentation Attack Detection. In Sébastien Marcel, Mark S. Nixon, and Stan Z. Li, editors, *Handbook of Biometric Anti-Spoofing*, pages 49–70. Springer, 2019.

[112] Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. CNN Fixations: An Unraveling Approach to Visualize the Discriminative Image Regions. *IEEE Transactions on Image Processing*, 28(5):2116–2125, 2018.

[113] Valerio Mura, Luca Ghiani, Gian Luca Marcialis, Fabio Roli, David A Yambay, and Stephanie A Schuckers. LivDet 2015 - Fingerprint Liveness Detection Competition 2015. In *Proc. IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6, 2015.

[114] Valerio Mura, Giulia Orrù, Roberto Casula, Alessandra Sibiriu, Giulia Loi, Pierluigi Tuveri, Luca Ghiani, and Gian Luca Marcialis. LivDet 2017 Fingerprint Liveness Detection Competition 2017. In *Proc. IAPR International Conference on Biometrics (ICB)*, pages 297–302, 2018.

[115] Karthik Nandakumar and Anil K Jain. Biometric Template Protection: Bridging the Performance Gap Between Theory and Practice. *IEEE Signal Processing Magazine*, 32(5):88–100, 2015.

[116] Mohammad-Reza Nasiri-Avanaki et al. Anti-spoof Reliable Biometry of Fingerprints using *En-face* Optical Coherence Tomography. *Optics and Photonics Journal*, 1(03):91–96, 2011.

[117] ABC News. Surgically Altered Fingerprints Help Woman Evade Immigration, 2009. abcnews.go.com/Technology/GadgetGuide/surgically-altered-fingerprints-woman-evade-immigration/story?id=9302505.

[118] Dinh-Luan Nguyen, Kai Cao, and Anil K Jain. Robust Minutiae Extractor: Integrating Deep Networks and Fingerprint Domain Knowledge. In *Proc. IAPR International Conference on Biometrics (ICB)*, pages 9–16, 2018.

[119] Rodrigo Frassetto Nogueira, Roberto de Alencar Lotufo, and Rubens Campos Machado. Fingerprint Liveness Detection Using Convolutional Neural Networks. *IEEE Transactions on Information Forensics and Security*, 11(6):1206–1213, 2016.

[120] Federal Bureau of Investigation. *The Science of Fingerprints: Classification and Uses, Rev 12-84.* U.S. Government Printing Office, Washington, DC, 1984.

[121] Federal Bureau of Investigation. Fbi warns about altered fingerprints. www.forensicmag.com/article/2015/05/fbi-warns-about-altered-fingerprints, 2015.

[122] Federal Bureau of Investigation. Fugitives on the FBI's 10 Most Wanted List. http://www.businessinsider.com/fbi-10-most-wanted-criminals-list-2017-11, 2018.

[123] Office of the Direction of National Intelligence (ODNI), IARPA. IARPA-BAA-16-04 (Thor). https://www.iarpa.gov/index.php/research-programs/odin/odin-baa, 2016.

[124] A OIG. Review of the FBI's Handling of the Brandon Mayfield Case. *Office of the Inspector General, Oversight and Review Division, US Department of Justice*, pages 1–330, 2006.

[125] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[126] Federico Pala and Bir Bhanu. Deep Triplet Embedding Representations for Liveness Detection. In *Deep Learning for Biometrics*, pages 287–307. Springer, 2017.

[127] 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.

[128] Sujan Parthasaradhi, Reza Derakhshani, Larry Hornak, and Stephanie Schuckers. Time-series Detection of Perspiration as a Liveness Test in Fingerprint Devices. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(3):335–343, 2005.

[129] George Pavlich. The emergence of habitual criminals in 19 th century britain: Implications for criminology. *Journal of Theoretical & Philosophical Criminology*, 2(1), 2010.

[130] Heinz-Helmut Perkampus. *UV-VIS Spectroscopy and its Applications*. Springer Science & Business Media, 2013.

[131] Richard Plesh, Keivan Bahmani, Ganghee Jang, David Yambay, Ken Brownlee, Timothy Swyka, Precise Biometrics, Peter Johnson, Arun Ross, and Stephanie Schuckers. Fingerprint Presentation Attack Detection utilizing Time-Series, Color Fingerprint Captures. In *IEEE International Conference on Biometrics (ICB)*, 2019.

[132] Carmen A Puliafito, Michael R Hee, Charles P Lin, Elias Reichel, Joel S Schuman, Jay S Duker, Joseph A Izatt, Eric A Swanson, and James G Fujimoto. Imaging of Macular Diseases With Optical Coherence Tomography. *Ophthalmology*, 102(2):217–229, 1995.

[133] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[134] Nalini K Ratha, Shaoyun Chen, and Anil K Jain. Adaptive Flow Orientation-based Feature Extraction in Fingerprint Images. *Pattern Recognition*, 28(11):1657–1672, 1995.

[135] Ajita Rattani, Walter J Scheirer, and Arun Ross. Open Set Fingerprint Spoof Detection Across Novel Fabrication Materials. *IEEE Transactions on Information Forensics and Security*, 10(11):2447–2460, 2015.

[136] Charles D. Robison and Maxwell S. Andrews. System and Method of Fingerprint Anti-spoofing Protection using Multi-spectral Optical Sensor Array, March 26 2019. US Patent 10,242,245.

[137] Arun Ross and Anil Jain. Biometric Sensor Interoperability: A Case Study in Fingerprints. In *International Workshop on Biometric Authentication*, pages 134–145. Springer, 2004.

[138] Robert K Rowe and David P Sidlauskas. Multispectral Biometric Sensor. US Patent 7,147,153, 2006.

[139] Proteek Chandan Roy and Vishnu Naresh Boddeti. Mitigating information leakage in image representations: A maximum entropy approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2586–2594, 2019.

[140] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet Large Scale Visual Recognition Challenge. *Proc. International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[141] S Sangiorgi, A Manelli, T Congiu, A Bini, G Pilato, M Reguzzoni, and M Raspanti. Microvascularization of the Human Digit as studied by Corrosion Casting. *Journal of Anatomy*, 204(2):123–131, 2004.

[142] Stephanie Schuckers and Peter Johnson. Fingerprint Pore Analysis for Liveness Detection, November 14 2017. US Patent 9,818,020.

[143] Study Scientific Working Group on Friction Ridge Analysis and Technology (SWGFAST). Standards for Examining Friction Ridge Impressions and Resulting Conclusions version 1.0, 2011.

[144] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[145] Nathan Silberman and Sergio Guadarrama. TensorFlow-Slim Image Classification Model Library. https://github.com/tensorflow/models/tree/master/research/slim.

[146] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034*, 2013.

[147] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[148] Brian C Smith. *Fundamentals of Fourier Transform Infrared Spectroscopy*. CRC press, 2011.

[149] Claron W Swonger, Dan M Bowers, and Robert M Stock. Fingerprint-based Access Control and Identification Apparatus. US Patent 4,210,899, 1980.

[150] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[151] Elham Tabassi. NIST Fingerprint Image Quality, NFIQ 2.0, 2016.

[152] Elham Tabassi, Tarang Chugh, Debayan Deb, and Anil K. Jain. Altered Fingerprints: Detection and Localization. In *IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9, 2018.

[153] Philippe Thévenaz, Thierry Blu, and Michael Unser. Image Interpolation and Resampling. *Handbook of Medical Imaging, Processing and Analysis*, 1(1):393–420, 2000.

[154] THORLabs. Telesto series (TEL1325LV2) Spectral-domain OCT scanner. https://www.thorlabs.com/catalogpages/Obsolete/2017/TEL1325LV2-BU.pdf, 2017.

[155] Michela Tiribuzi, Marco Pastorelli, Paolo Valigi, and Elisa Ricci. A multiple kernel learning framework for detecting altered fingerprints. In *21st International Conference on Pattern Recognition (ICPR)*, pages 3402–3405, 2012.

[156] Ruben Tolosana, Marta Gomez-Barrero, Christoph Busch, and Javier Ortega-Garcia. Biometric Presentation Attack Detection: Beyond the Visible Spectrum. *IEEE Transactions on Information Forensics and Security*, 2019.

[157] Mitchell Trauring. Automatic Comparison of Finger-ridge Patterns. *Nature*, 197(4871):938, 1963.

[158] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved Texture Networks: Maximizing Quality and Diversity in Feed-forward Stylization and Texture Synthesis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6924–6932, 2017.

[159] Unique Identification Authority of India: Govt. of India. Aadhaar Dashboard. https://uidai.gov.in/aadhaar_dashboard/, 2019.

[160] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5239–5247, 2017.

[161] C. Watson, G. Fiumara, E. Tabassi, S. L. Chang, P. Flanagan, and W. Salamon. Fingerprint Vendor Technology Evaluation (FpVTE). NIST Interagency Report 8034, 2015.

[162] Julia Welzel. Optical Coherence Tomography in Dermatology: A Review. *Skin Research and Technology: Review article*, 7(1):1–9, 2001.

[163] Kasey Wertheim. Embryology and Morphology of Friction Ridge Skin. *The Fingerprint Sourcebook*, pages 1–26, 2011.

[164] John J Wild and John M Reid. Application of Echo-ranging Techniques to the Determination of Structure of Biological Tissues. *Science*, 115(2983):226–230, 1952.

[165] Zhihua Xia, Chengsheng Yuan, Rui Lv, Xingming Sun, Neal N Xiong, and Yun-Qing Shi. A Novel Weber Local Binary Descriptor for Fingerprint Liveness Detection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[166] Wenqi Xian et al. TextureGAN: Controlling Deep Image Synthesis with Texture Patches. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8456–8465, 2018.

[167] David Yambay, Luca Ghiani, Paolo Denti, Gian Luca Marcialis, Fabio Roli, and S Schuckers. LivDet 2011-Fingerprint Liveness Detection Competition 2011. In *Proc. IAPR International Conference on Biometrics (ICB)*, pages 208–215, 2012.

[168] David Yambay, Luca Ghiani, Gian Luca Marcialis, Fabio Roli, and Stephanie Schuckers. Review of Fingerprint Presentation Attack Detection Competitions. In Sébastien Marcel, Mark S Nixon, Julian Fierrez, and Nicholas Evans, editors, *Handbook of Biometric Anti-Spoofing*. Springer, 2019.

[169] Wei-Yun Yau, Hoang-Thanh Tran, Eam-Khwang Teoh, and Jian-Gang Wang. Fake finger detection by finger color change analysis. In *International Conference on Biometrics*, pages 888–896. Springer, 2007.

[170] 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.

[171] Soweon Yoon and Anil K Jain. Longitudinal Study of Fingerprint Recognition. *Proceedings of the National Academy of Sciences*, 112(28):8555–8560, 2015.

[172] Yongliang Zhang, Daqiong Shi, Xiaosi Zhan, Di Cao, Keyi Zhu, and Zhiwei Li. Slim-ResCNN: A Deep Residual Convolutional Neural Network for Fingerprint Liveness Detection. *IEEE Access*, 7:91476–91487, 2019.

[173] Ding-Xuan Zhou. Universality of Deep Convolutional Neural Networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020.