Fingerprint Disentanglement for Presentation Attack Generalization Across Sensors and Materials

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

*Master of Science in **Electronics and Communication Engineering** by Research*

Gowri Lekshmy
20171053
gowri.lekshmy@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Fingerprint Disentanglement for Presentation Attack Generalization Across Sensors and Materials" by Gowri Lekshmy, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Prof. Anoop Namboodiri

To my family

Acknowledgments

I would like to express my heartfelt gratitude to my advisor Prof. Anoop Namboodiri for his guidance and support throughout my research program. I am grateful for the calm and friendly manner in which he engaged with me during our discussions, which made it easier to express my ideas. His mentorship gave me support and encouragement while helping me become an independent thinker. Further, I am thankful for his intuitive guidance, where he breaks down complex problems into manageable chunks, making them easier to conquer.

I thank my labmate Additya for his thoughtful suggestions and guidance in the early stage of my research.

I would also like to thank my CVIT labmates for helping and suggesting various ideas to solve and overcome challenges in my work. I also thank them for their reviews of my work.

I would like to extend my gratitude to all my professors and IIIT academics that gave me exposure to various evolving fields in technology, primarily artificial intelligence.

I also thank the lab and the institute for providing the computational resources and infrastructure required for my research.

I thank the staff and administration, especially Saketh, for promptly resolving any issues. I also thank them for providing me with the opportunity to travel and present my work at the conference.

I thank all my friends for supporting me and making my college journey enriching.

Finally, I am deeply grateful to my family for their unwavering support and for standing by me during the entirety of my college experience.

Abstract

In today's digital era, biometric authentication has become increasingly widespread for verifying a user across a range of applications, from unlocking a smartphone to securing high-end systems. Various biometric modalities such as fingerprint, face, and iris offer a distinct way to recognize a person automatically. Fingerprints are one of the most prevalent biometric modalities. They are widely utilized in security systems owing to their remarkable reliability, distinctiveness, invariance over time and user convenience.

Nowadays, automatic fingerprint recognition systems have become a prime target for attackers. Attackers fabricate fingerprints using materials like playdoh and gelatin, making it hard to distinguish them from live fingerprints. This way of circumventing biometric systems is called a presentation attack (PA). To identify such attacks, a PA detector is added to these systems.

Deep learning-based PA detectors require large amounts of data to distinguish PA fingerprints from live ones. However, there exists significantly less training data with novel sensors and materials. Due to this, PA detectors do not generalize well on introducing unknown sensors or materials. It is incredibly challenging to physically fabricate an extensive train dataset of high-quality counterfeit fingerprints generated with novel materials captured across multiple sensors. Existing fingerprint presentation attack detection (FPAD) solutions improve cross-sensor and cross-material generalization by utilizing style-transfer-based augmentation wrappers over a two-class PA classifier. These solutions generate large artificial datasets for training by using style transfer which learns the style properties from a few samples obtained from the attacker. They synthesize data by learning the style as a single entity, containing both sensor and material characteristics. However, these strategies necessitate learning the entire style upon adding a new sensor for an already known material or vice versa.

This thesis proposes a decomposition-based approach to improve cross-sensor and cross-material FPAD generalization. We model presentation attacks as a combination of two underlying components, i.e., material and sensor, rather than the entire style. By utilizing this approach, our method can generate synthetic patches upon introducing either a new sensor, a new material, or both. We perform two different methods of fingerprint factorization - traditional and deep-learning based. Traditional factorization of fingerprints into sensor and material representations using tensor decomposition establishes a baseline using machine learning for our hypothesis. The deep-learning method uses a decomposition-based augmentation wrapper for disentangling fingerprint style. The wrapper improves cross-sensor and cross-material FPAD, utilizing one fingerprint image of the target sensor and material. We also reduce

computational complexity by generating compact representations and utilizing lesser combinations of sensors and materials to produce several styles. Our approach enables us to generate a large variety of samples using a limited amount of data, which helps improve generalization.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Biometrics

Biometrics is derived from the Greek words - *bios* (life) and *metron* (measurement); biometric identifiers are measurements from a living human body.

Biometrics is the analysis of *unique anatomical characteristics*, such as fingerprints, iris, and face recognition, to identify a specific individual. These characteristics can be used to identify or verify a person's identity for security or other purposes.

Biometrics can be helpful in various contexts, including security, access control, and identification. Some specific examples of how biometrics can be used include:

- **Security:** Biometrics can be used to verify a person's identity before granting access to a secure area or system. This can help prevent unauthorized access and increase overall security.
- **Access control:** Biometrics can be used as a form of identification for access control systems, such as unlocking a smartphone or logging into a computer. This eliminates the need for traditional forms of identification, such as passwords or keys.
- **Identification:** Biometrics can be used to identify individuals in a variety of situations, such as border control, voting, and criminal investigations. This can help improve the accuracy and efficiency of identification processes.
- **Time and Attendance:** Biometric data can be used to track employee attendance and time spent on tasks, reducing the need for manual time cards or time sheets.
- **Banking and Finance:** Biometrics can be used to verify the identity of customers in banking and financial transactions, reducing the risk of fraud and increasing security.

Overall, biometrics can be a valuable tool for increasing security and convenience while making the identification processes more accurate and efficient.

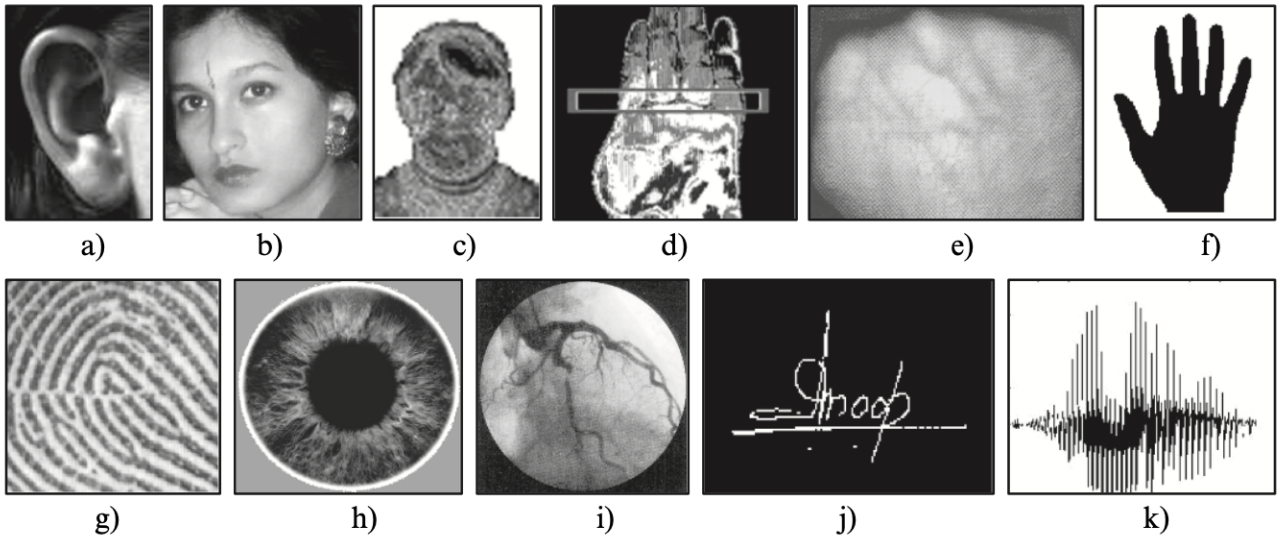


Figure 1.1: Different biometric modalities - a) ear, b) face, c) facial thermogram, d) hand thermo-gram, e) hand vein, f) hand geometry, g) fingerprint, h) iris, i) retina, j) signature, and k) voice [39]

To consider any anatomical or behavioral trait as a biometric identifier, the following conditions should be satisfied:

- **Universality:** every person must possess the trait.
- **Distinctiveness:** the trait should be sufficiently different for any two persons.
- **Permanence:** the trait should be invariant (with respect to the matching criterion) over time.
- **Collectability:** quantitatively measuring the trait should be possible.

Multiple biometric modalities satisfy the above conditions, however fingerprint biometrics is considered to be one of the most widely used [39] and accepted forms of biometrics due to its reliability and ease of use.

1.2 Fingerprint Recognition Systems

Fingerprint recognition systems use fingerprints as a means of identifying or verifying a person's identity. These systems typically consist of hardware and software that work together to capture, process, and analyze fingerprints. Initially, images of a person's fingerprints are captured, which are then analyzed using specific algorithms to extract unique feature points called minutiae. The minutiae are then compared to a database of fingerprints to find a match.

There are two main types of fingerprint recognition systems:

1. **AFIS (Automated Fingerprint Identification System)** systems are used for *identification*. These systems capture a person's fingerprints and compare them to a database of fingerprints to find a match. AFIS systems are commonly used in criminal investigations, border control, and other situations where identifying an individual is essential.

2. **ABIS (Automated Fingerprint Verification System)** systems are used for *verification*. These systems capture a person's fingerprints and compare them to a stored template or reference fingerprint to confirm their identity. ABIS systems are commonly used in access control, time and attendance, and other situations where verifying an individual's identity is important.

1.2.1 Fingerprint Presentation Attack Detection

Fingerprint Presentation Attack Detection (FPAD) is the process of identifying and preventing attempts to bypass fingerprint recognition systems using fraudulent means called presentation attacks (PA). The ISO standard *IEC 30107-1:2016(E)* [1] defines presentation attacks as the "*presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system*".

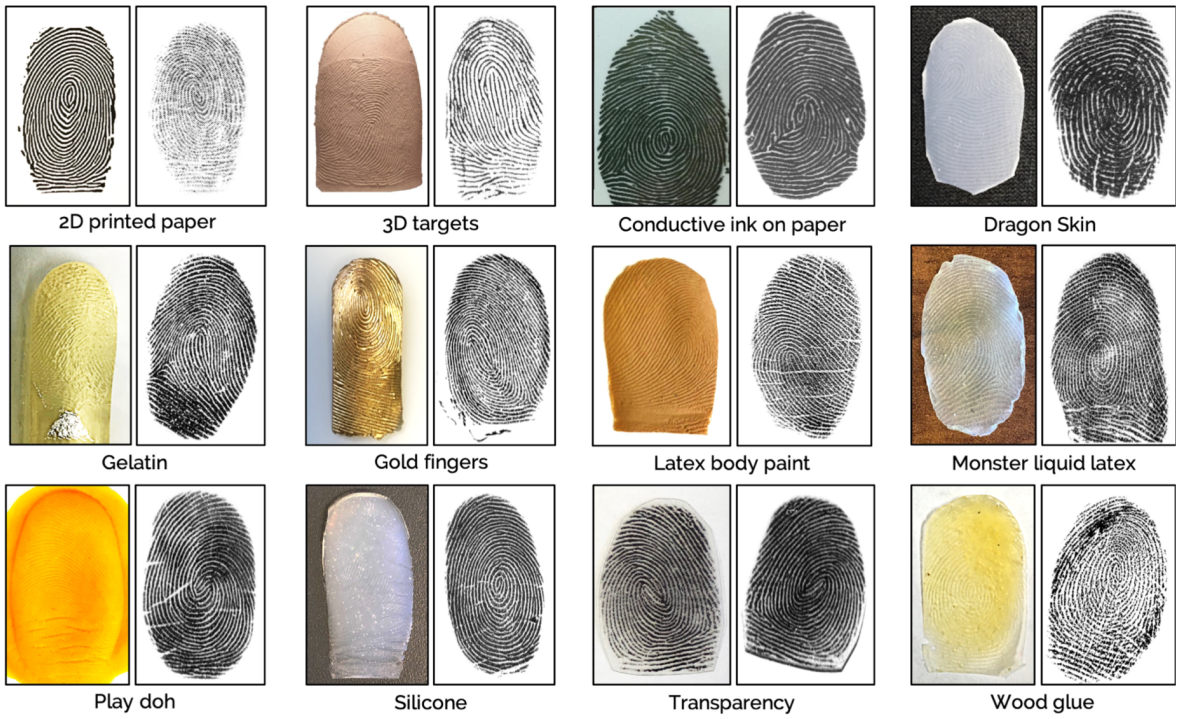


Figure 1.2: Fingerprint spoof examples from MSU 12 Spoof materials dataset [13]

Common presentation attacks include the usage of gummy [46] and spoof fingerprints created with readily available materials like playdoh, silicone, and gelatin. 2D or 3D printed fingerprint targets [3,

4, 6], altered fingerprints [63], and cadaver fingers [40] are a few sophisticated strategies to bypass fingerprint recognition systems.

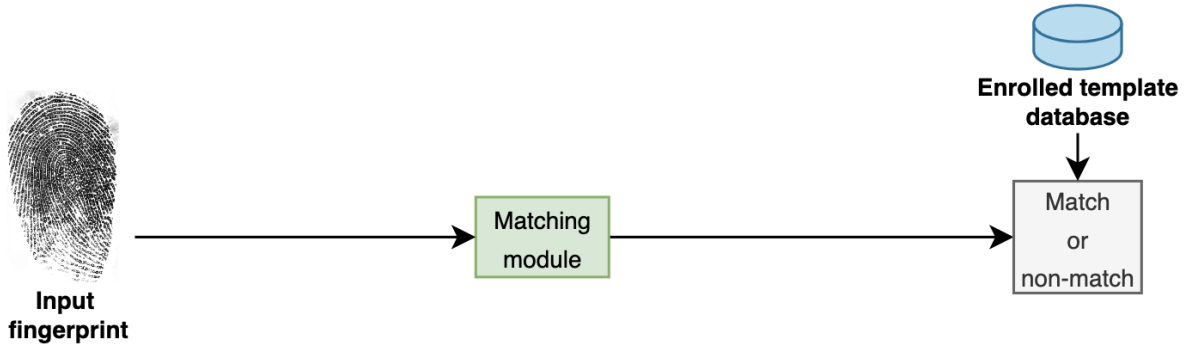


Figure 1.3: Fingerprint Recognition System vulnerable to presentation attacks.

The progressive demand for automatic fingerprint recognition systems has increased the number of presentation attacks. This poses a severe threat to automatic fingerprint recognition systems.

An ordinary automatic fingerprint recognition system as shown in Figure 1.3, uses a matching module to compare a query fingerprint with the other fingerprints in the database. Since this system is vulnerable to presentation attacks, the pipeline is modified to include a PA detector that can identify a spoof fingerprint, as shown in figure 1.4. If the fingerprint is detected as a non-spoof, it is sent to the verification or identification module.

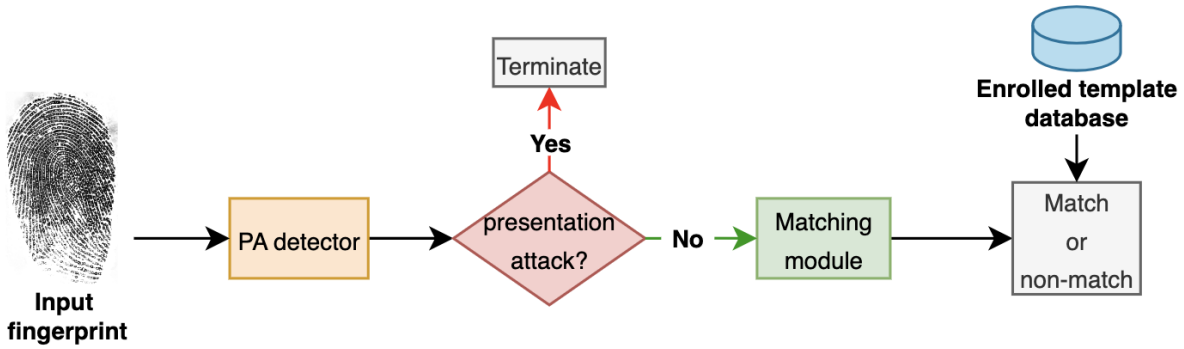


Figure 1.4: Fingerprint Recognition System with an additional presentation attack detector

1.2.2 Fingerprint Presentation Attack Generalization

Fingerprint presentation attack detectors are usually trained on datasets containing a limited set of live and spoof materials. However, attackers can bypass these systems using unknown spoof materials or

attack methods other than those encountered during training. Such attacks can be detected by improving the generalization capability of a fingerprint presentation attack detection system.

FPAD systems can be generalized using large and diverse datasets that cover a wide range of possible attacks. Retraining by augmenting synthetic datasets containing a variety of materials and sensors can also improve cross-sensor and cross-material generalization. In this thesis, we address the problem of FPAD generalization using a synthesis-based approach.

1.3 Motivation

Automatic fingerprint recognition systems are currently under the constant threat of presentation attacks (PAs). As mentioned in section 1.2.2, fingerprint presentation attack detection performance could be impacted due to the usage of an unknown sensor or material.

Some of the current concerns are the following:

1. Existing fingerprint presentation attack detection solutions improve cross-sensor and cross-material generalization by utilizing style-transfer-based augmentation wrappers over a two-class PAD classifier. These solutions synthesize data by learning the ***style as a single entity****, containing both sensor and material characteristics*. However, these strategies ***necessitate learning the entire style upon adding a new sensor for an already known material or vice versa***.

2. We might have *very few samples* from the attacker, serving as the target material and sensor for a synthesis-based FPAD generalization.

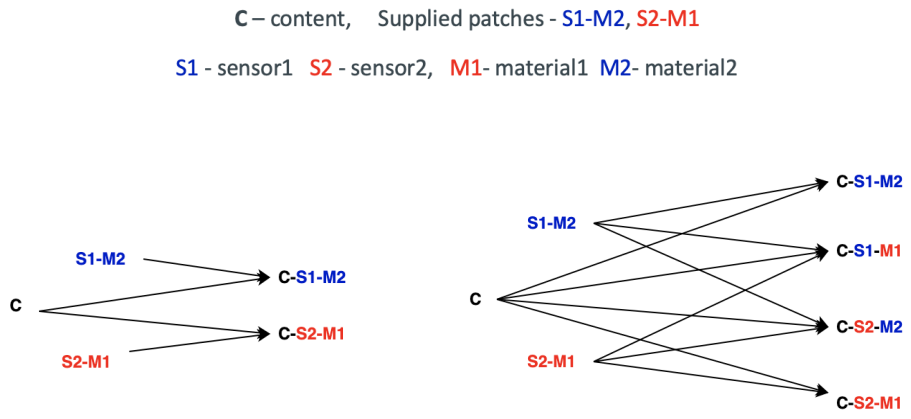
3. Existing works [13, 23] solve the problem by synthesizing fingerprints corresponding to unknown materials by *interpolation*. However, there is yet to be literature that generates data through extrapolation.

1.4 Contributions

The main contribution of this thesis is **Fingerprint Sensor and Material Disentanglement for FPAD Generalization**. We demonstrate two different methods of fingerprint factorization, traditional and deep-learning based, discussed in sections 1.4.1 and 1.4.2, respectively.

We explore a fundamentally new direction for modeling presentation attacks as a combination of two underlying components - material and sensor, rather than the entire style.

By utilizing a sensor and material decomposition-based approach, we can generate synthetic patches upon introducing a new sensor, material or both. Our method also reduces computational complexity by generating compact representations and utilizing lesser combinations of sensors and materials to produce several styles (Figure 1.5).



(a) Two output combinations using style transfer-based methods

(b) Four output combinations using our decomposition-based method

Figure 1.5: Consider supplied patches $S1$-$M2$ and $S2$-$M1$. Using style transfer methods as shown in (a), we can obtain only 2 output combinations. However, using our decomposition-based method, we can obtain 4 different output combinations as shown in (b). Here, C = content and $S1$, $S2$, $M1$, $M2$ = sensor1, sensor2, material1, material2

Generalization is commonly performed across different spoof materials, but we also perform generalization across sensors. Improving cross-sensor spoof detection performance is crucial to alleviate the time and resources involved in collecting large-scale datasets upon introducing new sensors. This is a highly-specific use case that can come to use in the case of a change of sensors used within large organizations for security or governments.

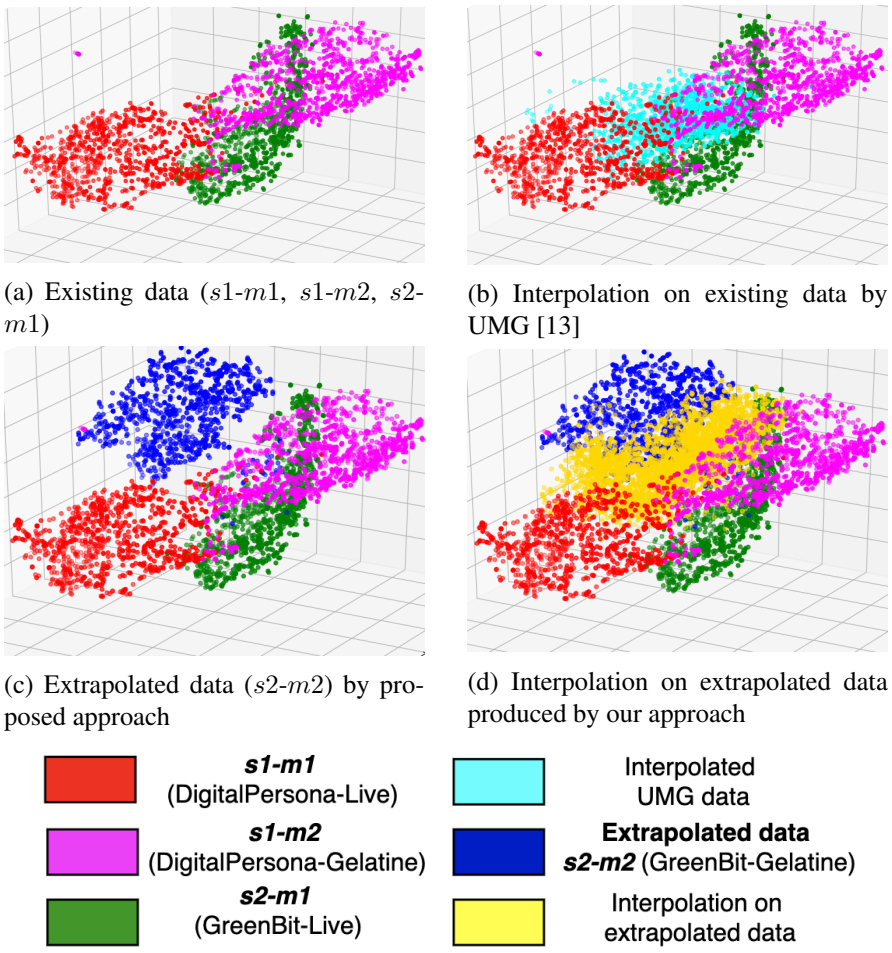
1.4.1 Traditional methods for Fingerprint Disentanglement

We factorize fingerprints using traditional methods to establish a baseline using machine learning. We use the N-mode SVD algorithm on a corpus of fingerprint data for decomposition and call it *tensorprints*. Tensorprints disentangle a fingerprint image into three different components - content (ridge-valley structure), sensor (the sensor used for capture) and material (live or spoof material texture). By fitting the tensor decomposition model on the train data, the algorithm learns to extract the underlying factors. The algorithm can then extrapolate or translate by generalizing to a new content, sensor or material. The learnt components are combined in multiple ways to obtain images across unknown sensor-material combinations. In this way, we synthesize images across unknown sensors and unknown materials.

1.4.2 Deep learning based Fingerprint Disentanglement

We propose a ***One-shot Sensor and Material Translator*** (OSMT) wrapper for improving cross-sensor and cross-material PAD. Our framework synthesizes large amounts of data across unknown sen-

sors and materials, from exclusively a *single fingerprint* by decomposing and combining the underlying sensor and material factors.



(a) Existing data ($s1$-$m1$, $s1$-$m2$, $s2$-$m1$)

(b) Interpolation on existing data by UMG [13]

(c) Extrapolated data ($s2$-$m2$) by proposed approach

(d) Interpolation on extrapolated data produced by our approach

Figure 1.6: 3D t-SNE visualization of style embeddings in the deep feature space produced by various methods of data generation - (a) Data for 3 existing sensor and material combinations, (b) Interpolated data between two existing sensor-material combinations by UMG, (c) Extrapolated data by our approach to obtain $s2$-$m2$ combination from existing data combinations ($s1$-$m1$, $s1$-$m2$, $s2$-$m1$) and (d) Further interpolation on our generated data. Here, $s1$, $s2$, $m1$, $m2$ = sensor1, sensor2, material1, material2

The main contributions of our work are as follows:

- To the best of our knowledge, this is the first work to disentangle a fingerprint image's style (textural characteristics) into its corresponding sensor and material embeddings. These embeddings are fused in multiple combinations to generate images across various unknown materials and sensors.

- For **n** sensors and **m** materials, our decomposition technique significantly reduces the number of style representations to be learned from **nxm** to just **n+m**.

- Disentanglement of sensor and material codes enables us to generate synthetic fingerprints by extrapolation. Further, we can generate even more data by interpolating between the extrapolated data and an existing material or sensor, as shown in Figure 1.6. This helps to develop data-augmentation approaches for robustness, generalizability and learning with small datasets.
- Our approach is a constrained learning problem. It produces *compact representations* of the sensor and material, which significantly helps in reducing the total number of parameters across all synthesized combinations.
- It uses only 50 minutiae patches from one fingerprint of an unknown material and sensor for synthesis, which is significantly low compared to the state-of-the-art [23], which utilizes atleast 100 live and 100 PA images to learn the target sensor.
- We present the improvement in PAD performance using our technique on the publicly available LivDet datasets (2015, 2017, 2019 and 2021).

1.5 Summary and Thesis Organization

The thesis is organized as follows:

- **Chapter 2** provides background on fingerprint presentation attack detection methods, techniques for FPAD generalization and representational learning for disentangling features.
- **Chapter 3** introduces traditional methods of factorization for fingerprints called *tensorprints*. It establishes the baseline using machine learning for our study on the disentanglement of fingerprint images.
- **Chapter 4** discusses the deep learning based approach for the disentanglement of fingerprint style and presents its superiority over the traditional factorization techniques. We also propose "One-shot Sensor and Material Translator" (OSMT) wrapper for improving cross-sensor and cross-material PAD.
- **Chapter 5** summarizes this thesis and presents ideas for future work.

Chapter 2

Literature Review

2.1 Fingerprint Presentation Attack Detection Techniques

Fingerprint presentation attack detection techniques can be based on hardware, software, or integrated solutions [43]. A combination of hardware and software-based solutions can provide additional layers of security and protection against presentation attacks.

2.1.1 Hardware-based Techniques

Hardware solutions are typically accomplished by using specialized sensors like OCT-based [10] or multi-spectral Lumidigm sensors. RaspiReader [16], an open-source fingerprint reader, uses two cameras to provide complementary streams (direct-view and FTIR) while capturing fingerprint images, which are both beneficial for spoof detection. These solutions also try to augment sensors to detect liveness with thermal output, odor [5] and blood flow [34].

Hardware-based methods can detect a wide range of spoofing techniques with good accuracy. However, they can also be expensive and require specialized equipment, limiting their applicability in certain settings. Additionally, some types of spoofing attacks may still be able to bypass these methods.

2.1.2 Software-based Techniques

On the other hand, software-based solutions extract features from the fingerprint image acquired by the sensor, to differentiate between live and spoof fingers. Software-based methods do not require specialized hardware, making them more cost-effective. These solutions can also be easily updated and deployed without any hardware changes.

2.1.2.1 Traditional FPAD Methods

Early software techniques used traditional methods to extract various handcrafted features from the fingerprint image and utilize them to classify it as either spoof or live.

[44] use 3rd-level anatomical features such as pore locations and their distributions for liveness detection. Some solutions exploit morphological [9] and perspiration-based characteristics [2] separately. Marasco *et al.* [42] combined a set of robust morphological and perspiration-based features for training different machine learning classifiers such as Support Vector Machine, Decision Tree, Multilayer Perceptron and Bayesian classifier.

Traditional classifiers also relied on texture based features such as Binarized Statistical Image Features (BSIF) [20], Weber Local Descriptor [22], and Local Phase Quantization (LPQ) [21]. LPQ method is renowned for being insensitive to blurring effects, thereby helping detect the differences between a live and a fake fingerprint due to the loss of information that may happen during the fabrication process. Drawing inspiration from [29] for face recognition and texture classification, Ghiani *et al.* proposed a novel fingerprint liveness descriptor named BSIF. BSIF automatically learns a fixed set of filters from a small set of natural images instead of using handcrafted filters, e.g., LBP and LPQ.

Rattani et al. [52] utilized Weibull-calibrated SVM (W-SVM) as a novel-material detector and a spoof detector, along with the capability for open set detection. [15] trained an ensemble of one-class SVMs on different feature sets, extracted only from live fingerprints to form a hypersphere.

2.1.2.2 Deep learning-based FPAD Methods

Recently, deep learning-based methods have shown to outperform traditional methods, achieving higher accuracy rates. Convolutional neural networks (CNNs) or other deep learning architectures can extract features automatically from the input images. They can also learn from raw data with less manual feature engineering than traditional methods, thereby motivating the incorporation of deep learning in biometrics.

A preliminary deep learning-based PA detector by Nogueira *et al.* [49] used a VGG for feature extraction and classification - both pre-trained and fine-tuned on fingerprints. Fingerprint images used by Nogueira*et al.* had lower regions of interest with white spaces. To resolve this issue, Pala *et al.* [51] used randomly cropped patches and trained their network with triplet loss. Chugh *et al.* [12] extracts localized minutiae patches aligned using fingerprint minutiae to provide salient cues for training a two-class PA classifier as shown in figure 2.1. SlimResCNN [65], the winner of LivDet 2017 [48], proposed a lightweight CNN with lesser processing time.

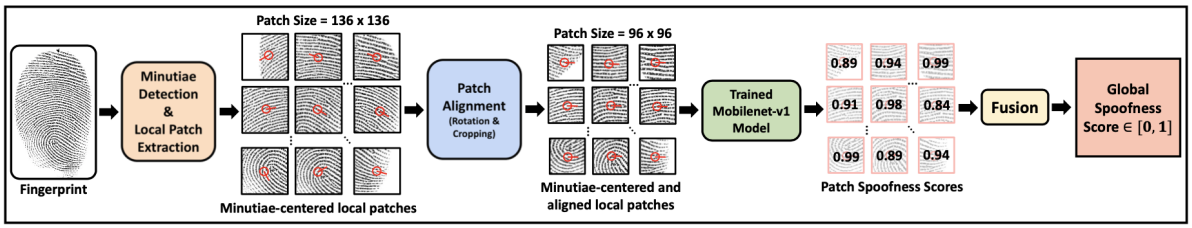


Figure 2.1: Overview of CNN-based Fingerprint Spoof buster for Fingerprint presentation attack detection [12]

2.2 Fingerprint Presentation Attack Detection Generalization

Published works [13, 18, 23] use style transfer-based augmentation wrappers over a PA detector to improve presentation attack detection performance on spoof materials with insufficient data.

Several style transfer works generate images with textures that are unseen during training. Gatys *et al.* [19] utilized CNNs for neural style transfer to generate stylized images by optimizing a noise image iteratively through a forward and backward pass. However, this is a computationally expensive optimization problem, so feed-forward network methods [28, 57] were used to find approximate solutions quickly. Another set of works [26, 58] utilizes feature statistics to perform style transfer.

[18] proposed the addition of synthesized spoof fingerprint patches by style transfer, while training the classifier. One limitation of this approach is that we have to re-train the classifier each time after adding synthesized patches for a new material to the train data.

To address this issue, UMG [13] synthesizes patches with style characteristics potentially similar to unknown spoof materials by interpolating the styles between known spoof materials. [23] incorporates adversarial representation learning on top of the UMG approach to improve cross-sensor generalization in addition to cross-material performance. However, our work in this thesis is distinct from the above approaches as we synthesize patches by extrapolation to enhance the generalization performance on both sensors and materials outside the convex hull of [13].

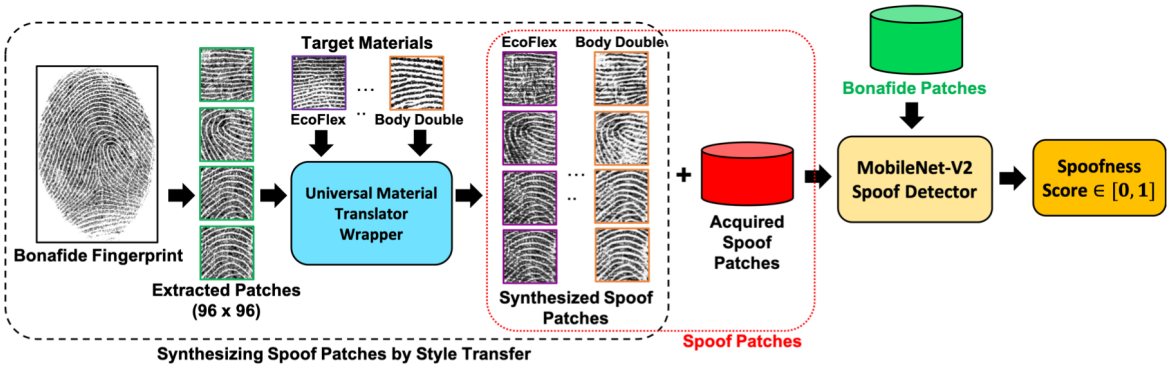


Figure 2.2: Spoof detector with a style transfer wrapper (UMT [18]) to prevent presentation attacks.

2.3 Disentangled Representation Learning

2.3.1 Traditional Factorization Models

Data is formed by multiple constituent factors along with interactions between them. Tenenbaum and Freeman [54] proposed that any entity can be separated into two factors - "style" and "content". [17, 55] put forth bilinear models for factorizing style and content using matrix decomposition. These bilinear frameworks can estimate the style and content vectors, and parameters independent of the style and content, but control their interaction. They use two approaches to fit the model - symmetric and

asymmetric. As shown in figure 2.3, this content-style factorization approach was used to solve three different problems - (i) *classification* of content with a new style, (ii) *extrapolation* of a new style to unseen content, and (iii) *translation* of new content observed in a new style. They applied bilinear factorization to data like typography, speech, and face-illumination as depicted in figure 2.4.

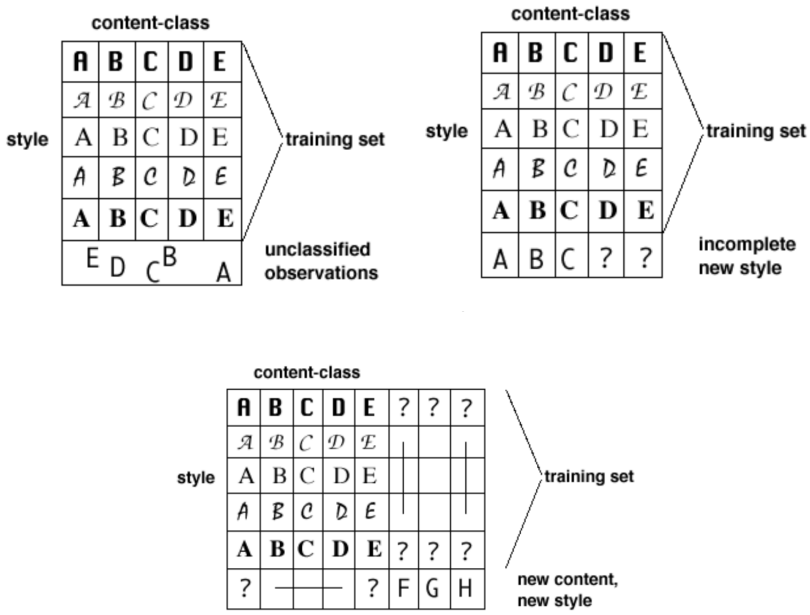


Figure 2.3: Given a training set of observations with multiple styles (i.e., fonts) and content classes (i.e., letters), we can (A) classify content with a new style, (B) extrapolate a new style to unobserved content classes, and (C) translate from new content observed only in new styles into known content or styles classes. [17]

These studies led to the development of multi-linear models analyzing data as a combination of two or more factors. Tucker [56] was the first to propose n-mode analysis. The N-mode SVD is a tensor extension of the conventional matrix singular value decomposition (SVD) employed for multi-linear modeling. Vasilescu *et al.* exploits the N-mode SVD algorithm in various domains, such as face recognition [61] and human motion synthesis [59]. First, [59] performed a 3-mode analysis breaking down the human motion tensor into people, action and joint angle time samples. Extracting these elements and recombining them in various ways yields a generative model. It can also recognize people and actions. Later, Vasilescu *et al.* decomposes faces into more than three factors (people, views, illumination and expressions), calling them "*tensorfaces*" [60].

2.3.2 Deep-Learning based Disentanglement

Existing works [33, 45, 62] achieve disentanglement of images into multiple factors of variation. Another set of works [31, 64] focuses on content-style separation by decomposing into the two factors. [64] uses EMD to generate images with unknown style and content given a few reference images.

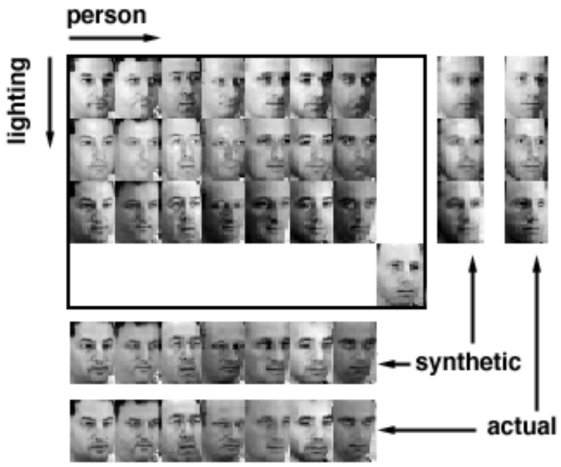


Figure 2.4: Example of translation on face images from [17]

The style and content encoders extract style and content representations. A bilinear mixer mixes these representations to generate images with target styles and contents. Lin *et al.* [36] uses bilinear CNN models for fine-grained categorization, which extract two features and combine them to obtain an image descriptor. To combine them, the outputs are multiplied using an outer product that can model pairwise interactions and later pooled. In this thesis, we make use of this matrix outer product for combining sensor and material codes.

2.4 Few-Shot Image-to-Image Translation

Image-to-Image translation models such as [27, 37] translate images among seen classes and generate poor translation outputs if few images are given at training time. [38] can learn previously unseen classes given at test time through a few example images. In the fingerprint domain, [18] requires 150 target material spoof patches to extract the style of a novel material. Learning style characteristics from the fingerprint of the target sensor and material obtained from the attacker, from as few images as possible is crucial.

2.5 Summary

In this chapter, we explored some of the existing work on fingerprint presentation attack detection, both hardware and software solutions. Further, we focus on software techniques that use classical machine learning and deep learning-based methods. Since these solutions do not generalize well for variations in fingerprint capture (e.g., unfamiliar sensors or materials), we also review existing literature related to FPAD generalization.

Additionally, we dive into the disentanglement strategies based on tensor factorization and autoencoders. The following chapters discuss classical and deep-learning-based approaches for FPAD generalization with the introduction of sensor-material factorization. In the next chapter, we focus on traditional tensor decomposition for fingerprints, a domain yet to be explored.

Chapter 3

Classical Factorization Methods on Fingerprints

We introduce traditional methods of factorization for fingerprints in this chapter. This chapter establishes the *baseline using machine learning* for our study on the disentanglement of fingerprint images. Here, we present ***classical techniques for fingerprint image synthesis through decomposition and translation***.

3.1 Introduction

Natural images are formed by the combination of multiple independent factors. For example, we observe that facial images consist of several modes of variation, such as different facial geometries (people), head poses, expressions and lighting conditions [60]. Similarly, fingerprint images also consist of various components, namely the ridge-valley structure (content denoting the person), pressure, the sensor used for capture, fingerprint dryness and live or spoof material texture.

To understand the composition of an image, we can extract its disentangled factors. We achieve this by formulating the problem using tensor decomposition. By learning the components, we can perform generalization tasks such as extrapolation and translation.

This chapter mainly discusses the traditional factorization methods for fingerprints using **multi-linear** models.

The chapter is structured as follows: section 3.2 introduces tensor algebra terminology. Section 3.3 describes the N-mode SVD algorithm for multi-linear analysis and tensor decomposition. Section 3.4 describes tensorprints obtained by applying the N-mode SVD algorithm on a corpus of fingerprint data.

3.2 Tensor Algebra Terminology

In this section, we introduce the basic definitions of multi-linear algebra. We follow the notations from [60]. Scalars are denoted by lower case letters $(a, b, \ldots)$, vectors by bold lower case letters $(\mathbf{a}, \mathbf{b} \ldots)$, matrices by bold upper-case letters (A, B ...), and higher-order tensors by calligraphic upper-case letters $(\mathcal{A}, B \ldots)$.

A tensor is a multidimensional matrix, n-way array, or n-mode matrix. It is a higher order generalization of a first order tensor (vector) and a second order tensor (matrix). The order of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is N. We denote an element of $\mathcal{A}$ as $\mathcal{A}_{i_1 \ldots i_n \ldots i_N}$ or $a_{i_1 \ldots i_n \ldots i_N}$, where $1 \leq i_n \leq I_n$.

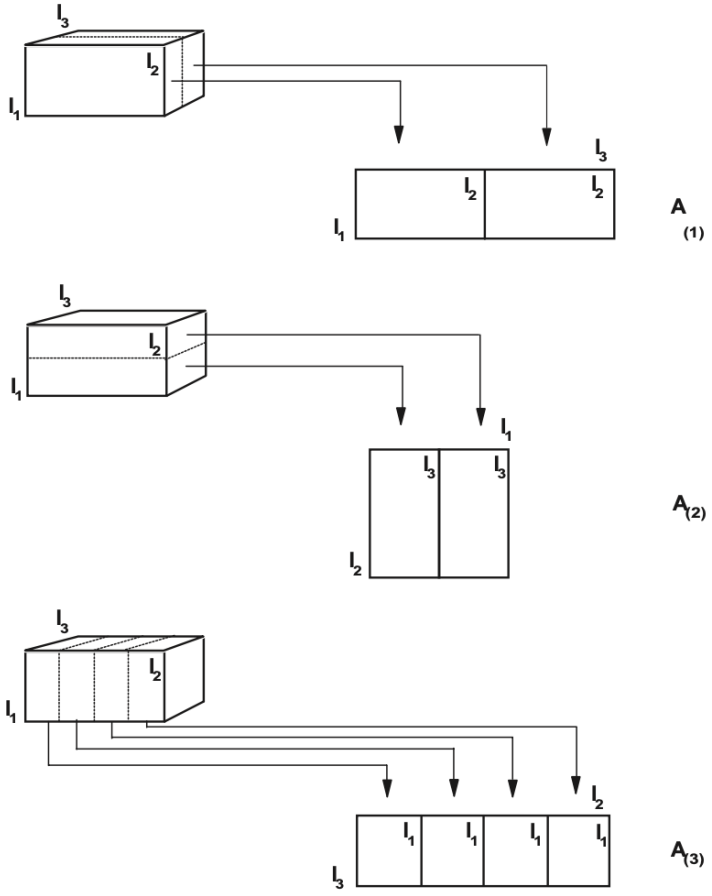


Figure 3.1: $\mathcal{A}_{(1)}, \mathcal{A}_{(2)}, \mathcal{A}_{(3)}$ matrices comprising mode-1, mode-2, and mode-3 vectors, obtained by flattening a 3rd-order tensor in 3 ways. [60]

Some of the important terms related to n-mode analysis for this thesis are as follows:

- mode-n vector: Consider an N^{th} order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, by varying index i_n while keeping the other indices fixed, we obtain I_n-dimensional vectors from $\mathcal{A}$. These vectors are called mode-n vectors. As shown in Figure 3.1, flattening the tensor $\mathcal{A}$ gives mode- n vectors that are the column vectors of matrix $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_N)}$. By generalizing for matrices, column vectors are referred to as mode-1 vectors and row vectors as mode-2 vectors.

- mode-n product: On generalizing the product of two matrices to a higher dimension, we require a product of a tensor and a matrix. The mode-n product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n \times \ldots \times I_N}$ by a matrix $\mathbf{M} \in \mathbb{R}^{J_n \times I_n}$ is denoted by $\mathcal{A} \times_n \mathbf{M}$. Let the resultant tensor be $\mathcal{B} \in \mathbb{R}^{I_1 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N}$,

whose entries are computed by

$$\mathcal{B}_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = (\mathcal{A} \times {}_n\mathbf{M})_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} m_{j_n i_n}. \tag{3.1}$$

The mode- n product in tensor notation is expressed as follows:

$$\mathcal{B} = \mathcal{A} \times {}_n\mathbf{M}, \tag{3.2}$$

or, in terms of flattened matrices,

$$\mathbf{B}_{(n)} = \mathbf{M}\mathbf{A}_{(n)}. \tag{3.3}$$

3.3 Tensor Decomposition - The N-Mode SVD Algorithm

Consider a matrix $\mathbf{D} \in \mathbb{R}^{I_1 \times I_2}$, a two-mode mathematical entity having two associated vector spaces - a row space and a column space.

The SVD algorithm can orthogonalize these two spaces and decompose the matrix into a product as follows:

$$\mathbf{D} = \mathbf{U}_1 \mathbf{\Sigma} \mathbf{U}_2^T \tag{3.4}$$

where the left matrix $\mathbf{U}_1 \in \mathbb{R}^{I_1 \times J_1}$ represents the orthogonal columnspace, $\mathbf{\Sigma} \in \mathbb{R}^{J_1 \times J_2}$ is a diagonal singular value matrix, and the orthogonal row space is represented by the right matrix $\mathbf{U}_2 \in \mathbb{R}^{I_2 \times J_2}$. Equation 3.4 can be rewritten in the form of mode-n product discussed in section 3.2.

$$\mathbf{D} = \mathbf{\Sigma} \times {}_1\mathbf{U}_1 \times {}_2\mathbf{U}_2 \tag{3.5}$$

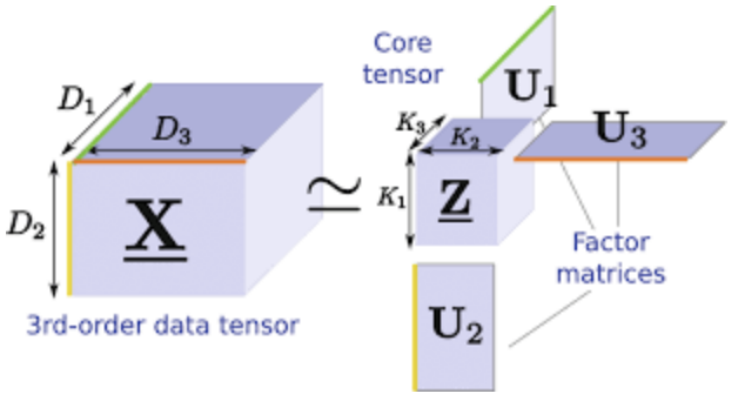


Figure 3.2: An N-mode SVD decomposes a tensor into N orthogonal spaces (vector spaces for N=3 shown above) from [24].

On extending to dimensions beyond two, we consider $\mathcal{D}$, an order $N > 2$ tensor comprising N spaces. The "N-mode SVD" for N-dimensional matrix orthogonalizes N spaces and expresses the tensor as an n-mode product of N-orthogonal spaces:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_n \mathbf{U}_n \ldots \times \times_N \mathbf{U}_N \tag{3.6}$$

In Equation 3.6, tensor $\mathcal{Z}$ is the core tensor and $\mathbf{U}_n$ are the mode matrices for $n = 1, \ldots, N$. The core tensor controls the interaction between all the mode matrices ($\mathbf{U}_n$). $\mathcal{Z}$ is similar to the diagonal singular value matrix in a conventional matrix SVD.

3.4 Tensor Prints

We perform factorization on fingerprints to get its constituent components. Here, we disentangle a fingerprint image into **three significant components** -

- Content (ridge-valley structure)
- Sensor (the sensor used for capture)
- Material (live or spoof material texture)

These components are later combined in multiple ways to obtain images across unknown sensor-material combinations.

Consider a fingerprint image data tensor $\mathcal{D}$ of the form $\mathbb{R}^{M \times N \times O \times P}$, where M corresponds to the number of sensors, N is the number of materials, O is the number of different content patches used, and P is number of pixels per image (flattened vector).

We apply the N-mode SVD algorithm from the above section 3.3 to decompose the tensor $\mathcal{D}$ into its corresponding core tensor $\mathcal{Z}$ and four orthogonal matrices - content $\mathbf{C}$, sensor $\mathbf{S}$, material $\mathbf{M}$ and pixels $\mathbf{P}$.

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{S} \times_2 \mathbf{M} \times_3 \mathbf{C} \times_4 \mathbf{P} \tag{3.7}$$

The sensor matrix $\mathbf{S} = [\mathbf{s}_1 \ldots \mathbf{s}_m \ldots \mathbf{s}_M]^T$, spans the space of sensor parameters, where each row vector $\mathbf{s}_m^T$ encodes the invariances for each sensor across different content and materials. Similarly, the material matrix $\mathbf{M} = [\mathbf{m}_1 \ldots \mathbf{m}_n \ldots \mathbf{m}_N]^T$, spans the space of content parameters, where each row vector $\mathbf{m}_n^T$ encodes the invariances for each material across different content and sensors. The content matrix $\mathbf{C} = [\mathbf{c}_1 \ldots \mathbf{c}_o \ldots \mathbf{c}_O]^T$, spans the space of content parameters, where each row vector $\mathbf{c}_o^T$ encodes the invariances for each content across different sensors and materials. The mode matrix $\mathbf{P}$ orthonormally spans the space of images.

We synthesis tensorprints using multi-linear-based factorization methods as written below in section 3.4.2.

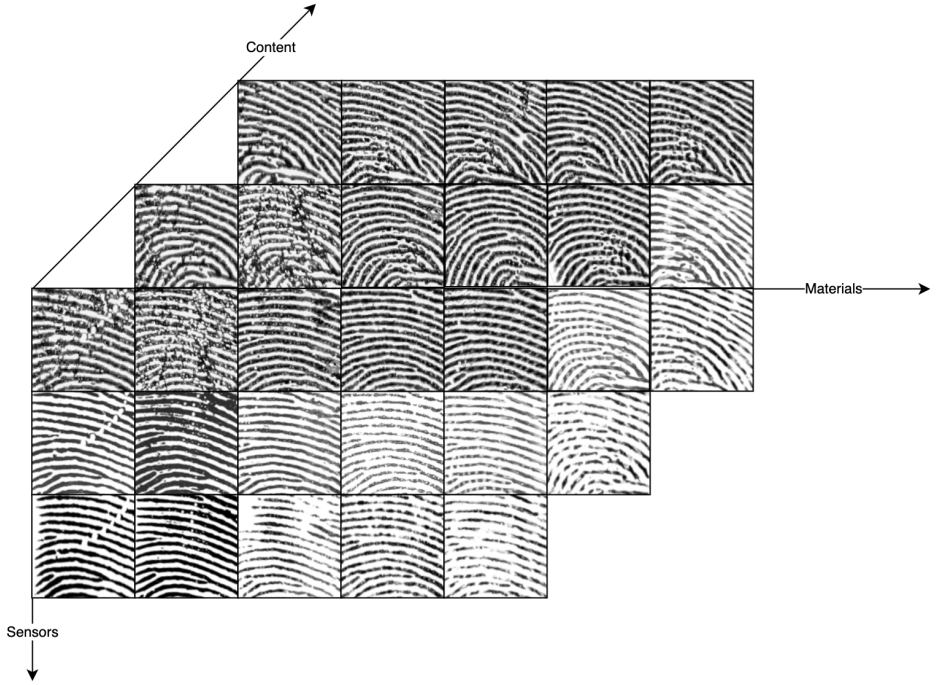


Figure 3.3: Data tensor with fingerprint variations across content, sensors and materials

3.4.1 Datasets

We utilize the fingerprint liveness dataset LivDet 2015 [47] for our analysis and synthesis. The images belong to 4 different optical sensors - Green Bit, Biometrika, Digital Persona and Crossmatch. The first three sensors (Green Bit, Biometrika, and Digital Persona) contain five materials in train - Ecoflex00-50, gelatine, latex, and wood glue and live. The CrossMatch sensor has materials that do not overlap with the materials of the other three sensors, so we exclude the sensor from our experiments. The total data has 3 sensors, 5 materials and 148 different content.

During training, we utilize the leave-one-out strategy and create images for the left-out sensor and material by translation. The data tensor used for factorization will $\mathcal{D} \in \mathbb{R}^{2\times4\times148\times9216}$. On decomposing $\mathcal{D}$ using equation 3.7, the core tensor $\mathcal{Z} \in \mathbb{R}^{2\times4\times148\times9216}$, $\mathbf{S} \in \mathbb{R}^{2\times2}$, $\mathbf{M} \in \mathbb{R}^{4\times4}$, $\mathbf{C} \in \mathbb{R}^{148\times148}$ and $\mathbf{P} \in \mathbb{R}^{9216\times9216}$.

Pre-processing

We first resize the 1000 dpi images of the Biometrika sensor in LivDet 2015 to 500 dpi. We preprocess these images by segmenting the fingerprint from the background similar to [18]. To create tensor $\mathcal{D}$, we must use patches with the same content across different sensors and materials. For this, we get 96x96 minutiae-aligned patches using the mated minutiae module of VeriFinger SDK v11 (commercial SDK). After aligning and cleaning the data by removing incomplete patches, we get $3 \times 5 \times 148 \times 9216$ as the size of the total data tensor.

Minutiae-based Alignment

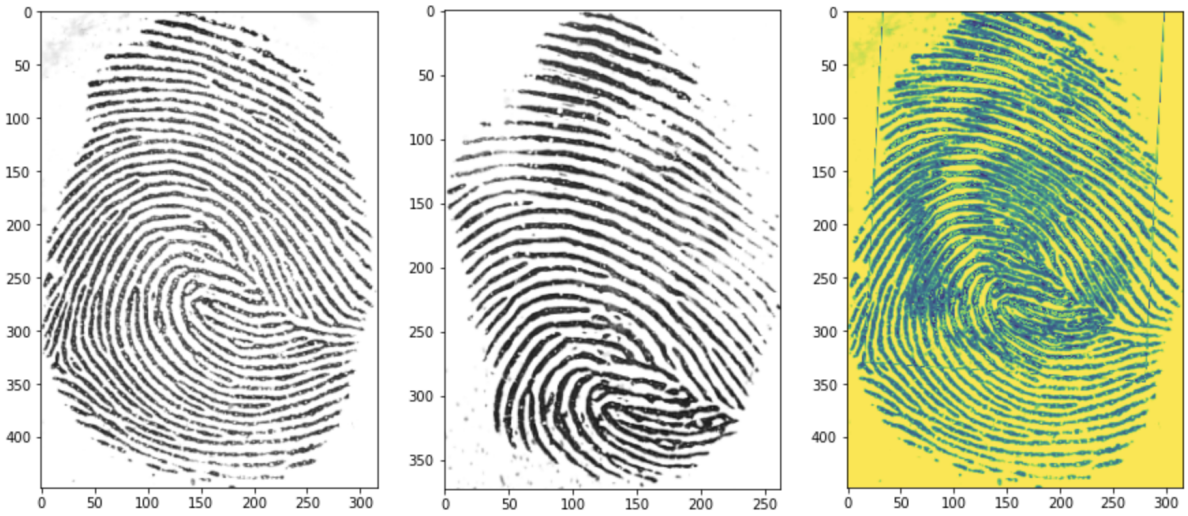
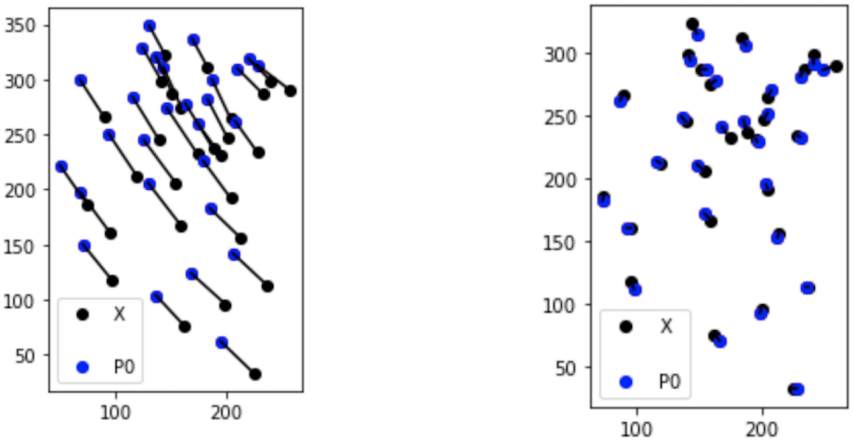


Figure 3.4: First two images correspond to the fingerprints to be aligned. Third image shows the overlapping images after performing alignment.

To form the data tensor, we need to use minutiae-aligned patches. We first gather the images belonging to the same fingerprint (content) but across all sensor-material variations - $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$. Then, we find the fingerprint with the maximum number of minutiae and set this as the reference fingerprint for alignment, denoted as $\mathbf{x}_r$. This will ensure the greatest overlap between all the impressions, and we will get the maximum number of aligned patches.



(a) Minutiae points before alignment

(b) Minutiae points after alignment

Figure 3.5: Minutiae points of the reference and gallery fingerprint before and after alignment

We then iterate through all the remaining impressions and find the corresponding mated minutiae with the reference fingerprint using the VeriFinger SDK v11. Let there be m_p matching minutiae pairs

between fingerprint $\mathbf{x}_p$ and $\mathbf{x}_r$, $\left\{(i_1, j_1), (i_2, j_2), \cdots, \left(i_{m_p}, j_{m_p}\right)\right\}$, where $p \neq r$. The matched minutiae pairs are the i_q-th minutia in fingerprint $\mathbf{x}_p$ and the j_q-th minutia in fingerprint $\mathbf{x}_r$, where $q \in \{1, 2, \cdots, m_p\}$. We find the affine transformation parameters $\mathbf{T}_{(p,r)}$ using the least squares method:

$$\mathbf{T}_{(p,r)} = \left(\left(\overline{\mathbf{P}}^{\mathrm{T}}\overline{\mathbf{P}}\right)^{-1}\overline{\mathbf{P}}^{\mathrm{T}}\mathbf{R}\right)^{\mathrm{T}}$$

where $\mathbf{P} \in \mathbb{R}^{m_p \times 2}, \overline{\mathbf{P}} = [\mathbf{P}, \mathbf{1}] \in \mathbb{R}^{m_p \times 3}$ and $\mathbf{R} \in \mathbb{R}^{m_p \times 2}$ are the matching minutiae pairs' coordinates in $\mathbf{x}_p$ and $\mathbf{x}_r$ respectively.

Using the obtained transformation matrix, we align all the fingerprint impressions to the reference fingerprint pairwise. After aligning, we crop patches around each minutiae point.

3.4.2 Linear Factorization

We perform linear factorization on fingerprints using the equation 3.7 to get the constituent components. We use the tensorly [30] library to perform tucker decomposition (or N-mode SVD).

Using multi-linear factorization, we analyze a corpus of fingerprint data spanning different sensors, materials and content. By fitting the linear decomposition model on the train data, the algorithm learns to extract the underlying factors well. The algorithm can then extrapolate or translate by generalizing to a new content, sensor or material. In this way, we can synthesize a large number of images for unknown sensors and unknown materials.

We train by utilizing the leave-one-out strategy with $\mathcal{D}$ and synthesis images for the left-out sensor and material by translation.

Consider $\mathcal{D}^{ijk} \in \mathbb{R}^{1\times1\times1\times P}$, a part of the tensor $\mathcal{D}$, which denotes the image vector for the i^{th} sensor, j^{th} material, and k^{th} content as follows:

$$\mathcal{D}^{ijk} = \mathcal{Z} \times {}_1{s_i}^T \times {}_2{m_j}^T \times {}_3{c_k}^T \tag{3.8}$$

Suppose we get two images, say $\mathbf{d}_{new}^{sensor}$ and $\mathbf{d}_{new}^{mat}$ from an attacker. $\mathbf{d}_{new}^{sensor}$ with an unknown sensor $\mathbf{s}_{M+1}$ and known material. $\mathbf{d}_{new}^{mat}$ with a known sensor and an unknown material $\mathbf{m}_{N+1}$.

We can use the N-mode SVD algorithm to split them into their linear components as below

$$\mathbf{d}_{new}^{sensor} = \mathcal{Z} \times {}_1(s_{M+1})^T \times {}_2(m_j)^T \times {}_3(c_k)^T \tag{3.9}$$

$$\mathbf{d}_{new}^{mat} = \mathcal{Z} \times {}_1(s_i)^T \times {}_2(m_{N+1})^T \times {}_3(c_k)^T \tag{3.10}$$

The tensors $\mathcal{B}_{sensor}$ and $\mathcal{B}_{mat}$ are calculated by leaving the mode matrices for sensor and material,

$$\mathcal{B}_{sensor} = \mathcal{Z} \times_2 (m_j)^T \times_3 (c_k)^T \tag{3.11}$$

$$\mathcal{B}_{mat} = \mathcal{Z} \times_1 (s_j)^T \times_3 (c_k)^T \tag{3.12}$$

We calculate $(s_{M+1})^T$ and $(m_{N+1})^T$ by the following equations 3.13 and 3.14 similar to [59, 35]. Here, $[\dots]^{-1}$ indicates the pseudo-inverse function.

$$\mathbf{d}_{new}^{sensor} = \mathcal{B}_{sensor} \times_1 (s_{M+1})^T, \mathbf{d}_{new}^{mat} = \mathcal{B}_{mat} \times_2 (m_{N+1})^T \tag{3.13}$$

$$(s_{M+1})^T = \mathbf{d}_{new}^{sensor} \left[\mathcal{B}_{sensor}\right]^{-1}, (m_{N+1})^T = \mathbf{d}_{new}^{mat} \left[\mathcal{B}_{mat}\right]^{-1} \tag{3.14}$$

To get the final synthesized image with the unknown sensor (s_{M+1}) and the unknown material (m_{N+1}) with any content (c_k), we use n-mode product as given in equation 3.15.

$$\mathbf{d}_{new}^{sensor,mat} = \mathcal{Z} \times_1 (s_{M+1})^T \times_2 (m_{N+1})^T \times_3 (c_k)^T \tag{3.15}$$

3.4.2.1 Experimental procedure

We utilize a PA classifier to evaluate the performance of our factorization method. We use a leave-one-out strategy for the particular test sensor and test material. The left-out train data is then augmented with the test data to increase the PA classifier's test size.

We synthesize the target sensor and material data as mentioned in 3.4.2. For capturing the style, we use fingerprints from the attacker and content from the bonafide patches of a known sensor. The PA detector is trained with and without these synthesized patches to verify the improvement.

3.4.2.2 Results

While evaluating cross-sensor and cross-material performance, we remove both the lives and spoof data of the unknown sensor from the train set. The data belonging to the novel material within all the other sensors in train data is also removed.

To generate patches of unknown sensor and unknown material, we supply two sets of 50 minutiae patches each from (i) unknown material and known sensor and (ii) known material and unknown sensor. We utilize 15,000 synthesized patches for every unknown sensor-material combination.

Figure 3.6 presents the synthesized patches generated by classical linear factorization.

Table 3.1 showcases the cross-sensor and cross-material performance obtained without and with synthesized patches (obtained by classical linear factorization).

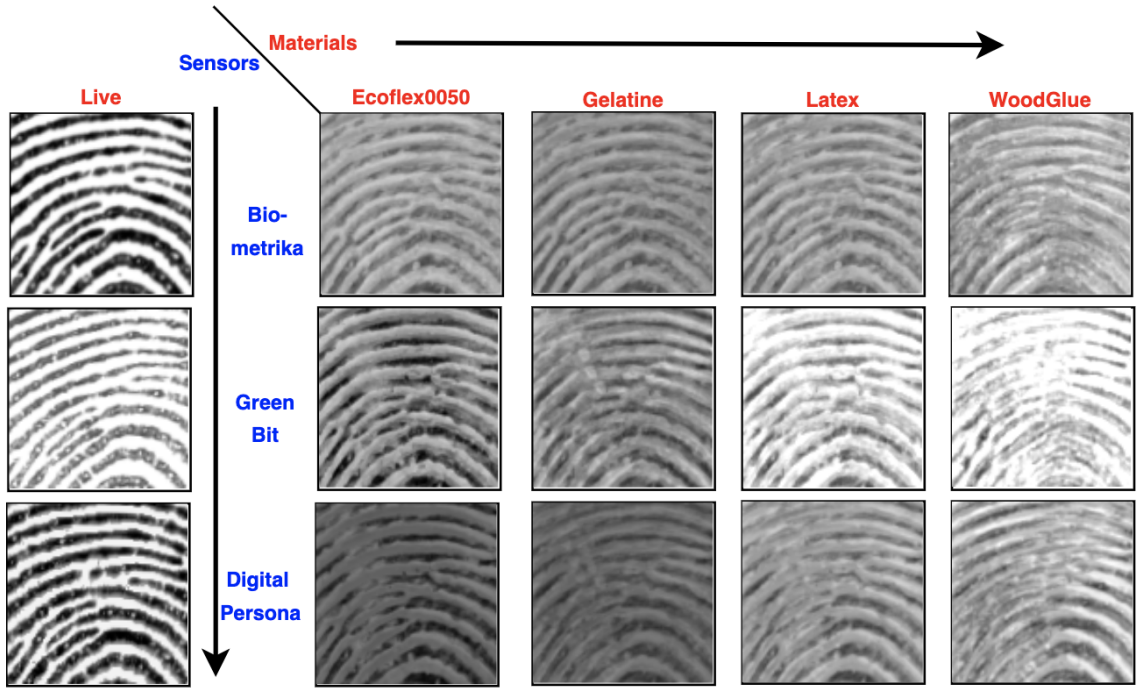


Figure 3.6: Synthesized patches for cross-sensor and cross-material combinations using classical linear method. Bonafide images for the corresponding sensors are in the first column.

Table 3.1: Cross-sensor and cross-material performance (TDR (%) @ FDR = 0.1%) without and with synthesized patches (obtained by classical linear N-mode SVD) of the corresponding test material and sensor in training

	GreenBit		**Digital Persona**		**Biometrika**	
Materials	**Without synthesized**	**With linear synthesized**	**Without synthesized**	**With linear synthesized**	**Without synthesized**	**With linear synthesized**
Ecoflex 00-50	96.07	75.67	10.46	6.57	47.34	51.51
Gelatine	81.52	81.82	4.37	2.34	21.74	20.73
WoodGlue	61.32	47.12	3.81	0.37	6.44	3.9
Latex	91.51	65.74	3.81	2.13	69.07	58.69

3.5 Summary

In this chapter, we established a baseline for fingerprint factorization using traditional machine-learning methods. We utilize multi-linear tensor decomposition (N-mode SVD algorithm) to find the corresponding content, sensor and material representations. We synthesize images with unknown sensors and materials using translation and feed these images to the PA detector for re-training. It is observed that the synthesized images could be of better quality. The cross-sensor and cross-material performance is poor, mainly due to the lack of training data, as tensorprints require aligned patches across all combinations. This motivates us to switch to deep learning-based methods, as introduced in the next chapter.

Chapter 4

Deep Learning-Based Disentanglement of Fingerprint Style

In this chapter, we discuss the ***deep learning approach for the disentanglement of fingerprint style*** and present its *superiority over the traditional factorization techniques* mentioned in the previous chapter.

4.1 Motivation for Deep Learning-Based Method

- **Captures Multi-level context (Local and global region):** The content of a region depends on its immediate context as well as its larger context. Traditional methods try to represent the overall information of an image by giving uniform importance to each sub-region. However, we need to capture the information such that the local region is given higher importance than the global context. This way, the style will be learned from the overall image, but the exact variations will depend on the local context. This kind of focused approach to the local region is required to enhance the feature representation. Deep learning provides a natural way to solve this problem using convolutional layers as shown in Figure 4.1.

- **Controllable Non-linearity:** Deep learning and kernel-based approaches both learn from the data. However, the kernel-based method captures non-linearity of a fixed nature depending on the specific kernel used, whereas deep learning is more flexible regarding its capabilities. Deep learning has the ability to model non-linearity from a simple level to a very complex degree. The non-linear complexity will be more for deep networks if they have multiple layers. Depending on the amount of data, if you use the right amount of regularization, neural networks will create representations as simple as possible but sufficiently capturing enough non-linearity to model it. It will keep iterating until the loss function becomes good but captures the underlying non-linearity completely. Therefore, deep learning helps in controllable non-linearity.

- **Ability to learn from more data:** Generally, traditional factorization-based methods find it difficult to deal with a large quantity of data. Deep learning-based techniques use neural networks with gradient descent and can iteratively learn using mini-batches on more data.

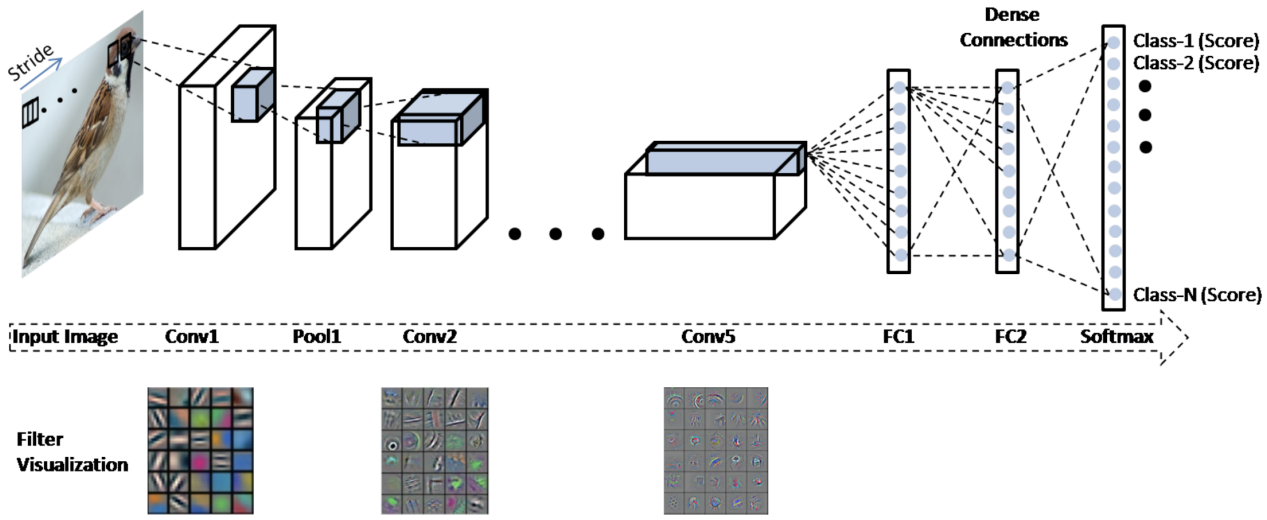


Figure 4.1: Convolutional Neural networks for capturing multi-level context [32]

4.2 Introduction

The increasing number of fingerprint presentation attacks poses a severe threat to automatic fingerprint recognition systems. Fingerprint presentation attack detection techniques are incorporated into fingerprint recognition systems to combat these attacks. State-of-the-art presentation attack[1] detectors using deep learning methods such as CNNs have demonstrated exceptionally high accuracy. However, CNN-based PA detectors have to be trained with large amounts of data captured across each spoof material and sensor for better generalization.

Attackers are constantly trying to exploit diverse fabrication techniques to produce significantly distinct spoof materials to circumvent biometric systems. It is observed that the PAD performance reduces significantly [41] upon introducing fingerprints synthesized using novel materials or those captured with unknown sensors. However, it is incredibly challenging to physically fabricate an extensive train dataset of high-quality counterfeit fingerprints generated with novel materials captured across multiple sensors.

Furthermore, we might barely have a few samples of a spoof material or sensor from the attacker. For example, cases like partial PA fingerprints captured on an unknown sensor can contribute extremely few patches for serving as the target material and sensor for a synthesis-based FPAD. Therefore, it is crucial to learn the characteristics of these novel materials and sensors with the help of a minimal number of samples to improve the cross-material and cross-sensor generalization capability of fingerprint PA detectors.

Existing works use a style-transfer-based augmentation wrapper over a PA classifier to improve presentation attack detection. UMT [18] generates a large synthetic PA dataset by using atleast 150 patches

[1]Presentation attack detection (PAD) is also referred to as liveness detection or spoof detection, we use these terms interchangeably.

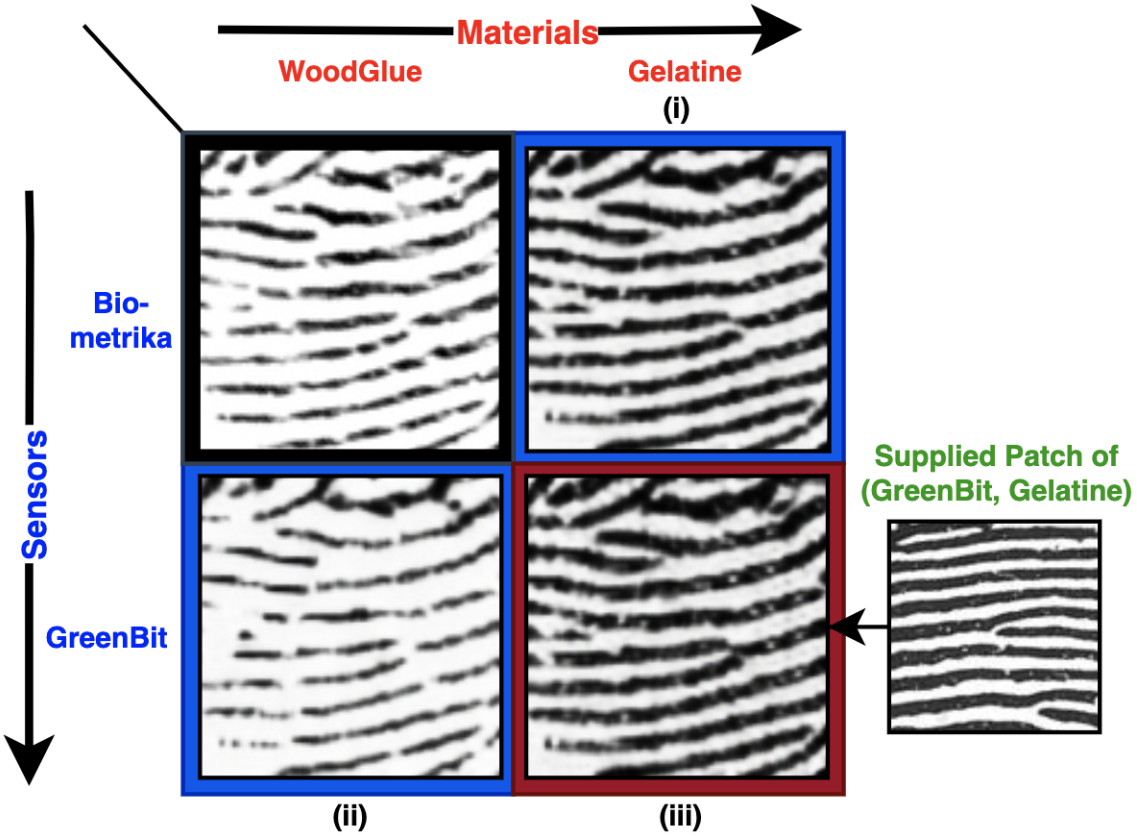


Figure 4.2: A supplied style patch is factorized into its corresponding sensor and material codes, then combined in several ways to synthesize patches across (i) unknown material and known sensor, (ii) known material and unknown sensor, or (iii) unknown material and unknown sensor. The black cell indicates an existing train image. From the supplied image's style, UMT [18] can only generate the style in the red cell, whereas our proposed method can generate the style in both the blue and red cells.

of an unknown material over a fixed CrossMatch sensor. Another set of works [13, 23] aims to improve cross-sensor and cross-material performance by synthesizing fingerprint images corresponding to novel materials, possibly occupying the space between the known materials in the deep feature space. However, these synthesized images only lie within the convex hull formed by interpolation between known materials which is a very limited and constrained feature space.

Our approach aims to improve performance by *extrapolating* to sensors and materials outside the convex hull, utilizing one fingerprint image of the target sensor and material. This allows us to explore a *much larger and diverse feature space, and generates a large variety of samples using the same limited amount of data, which helps improve generalization further.*

We propose a ***One-shot Sensor and Material Translator*** (OSMT) wrapper for improving cross-sensor and cross-material PAD. Our framework synthesizes large amounts of data across unknown sensors and materials, from exclusively a *single fingerprint* by decomposing and combining the underlying sensor and material factors.

4.3 Methodology

4.3.1 Overview

An outline of our end-to-end pipeline for improving presentation attack detection is depicted in Figure 4.3. We propose One-shot Sensor and Material Translator (OSMT) wrapper that disentangles material and sensor characteristics from a single target fingerprint and translates them onto the content (ridges) of other fingerprint images.

While training, we learn to translate images between known classes of sensors and materials. At test time, we extract sensor and material embeddings from only one fingerprint that belongs to an unknown material, sensor, or both. We utilize these embeddings to synthesize patches of the required target sensor-material combination by translation. These synthesized images are augmented to the existing train dataset of OSMT to train the MobileNet-V2 [53] PA classifier. This pipeline has proven to improve the FPAD performance for the target materials and sensors.

4.3.2 One-Shot Sensor and Material Translator

We input a 96×96 target patch of a novel material, sensor, or both, along with the required content fingerprint patch to the OSMT framework. We follow an architecture similar to Liu *et al.* [38], with a conditional image generator G and multi-task adversarial discriminator D. G can simultaneously take in a content patch $\mathbf{c}$, a set of $K1$ patches from a sensor $\mathbf{s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{K1}\}$ and $K2$ patches from a material $\mathbf{m} = \{\mathbf{m}_1, \ldots, \mathbf{m}_{K2}\}$ to produce an output patch $\overline{\mathbf{c}}$.

$$\overline{\mathbf{c}} = G\left(\mathbf{c}, \{\mathbf{s}, \mathbf{m}\}\right) = G\left(\mathbf{c}, \{\{\mathbf{s}_1, \ldots, \mathbf{s}_{K1}\}, \{\mathbf{m}_1, \ldots, \mathbf{m}_{K2}\}\}\right) \tag{4.1}$$

As shown in Figure 4.3, $\overline{\mathbf{c}}$ retains the content of $\mathbf{c}$ while resembling the textural characteristics of material $\mathbf{m}$ and sensor $\mathbf{s}$. We fix $K1 = K2 = 1$ in our experiments as we propose to extract the sensor and material codes from exclusively a single target patch.

4.3.2.1 Architecture

Our generator consists of a content encoder E_c, a material encoder E_m, a sensor encoder E_s, a bilinear layer B and a joint decoder F_c.

- **Content encoder:** Our content encoder E_c maps an input content fingerprint patch $\mathbf{c}$ to a content latent code $\mathbf{z}_c$, which is a spatial feature map. It comprises several strided convolutional layers for downsampling the input, followed by residual blocks [25]. Every convolutional layer is followed by Instance Normalization (IN) [58].

- **Sensor and Material encoders:** We pass the input style patch belonging to the desired sensor and material through the encoders E_s and E_m to extract corresponding sensor and material codes by

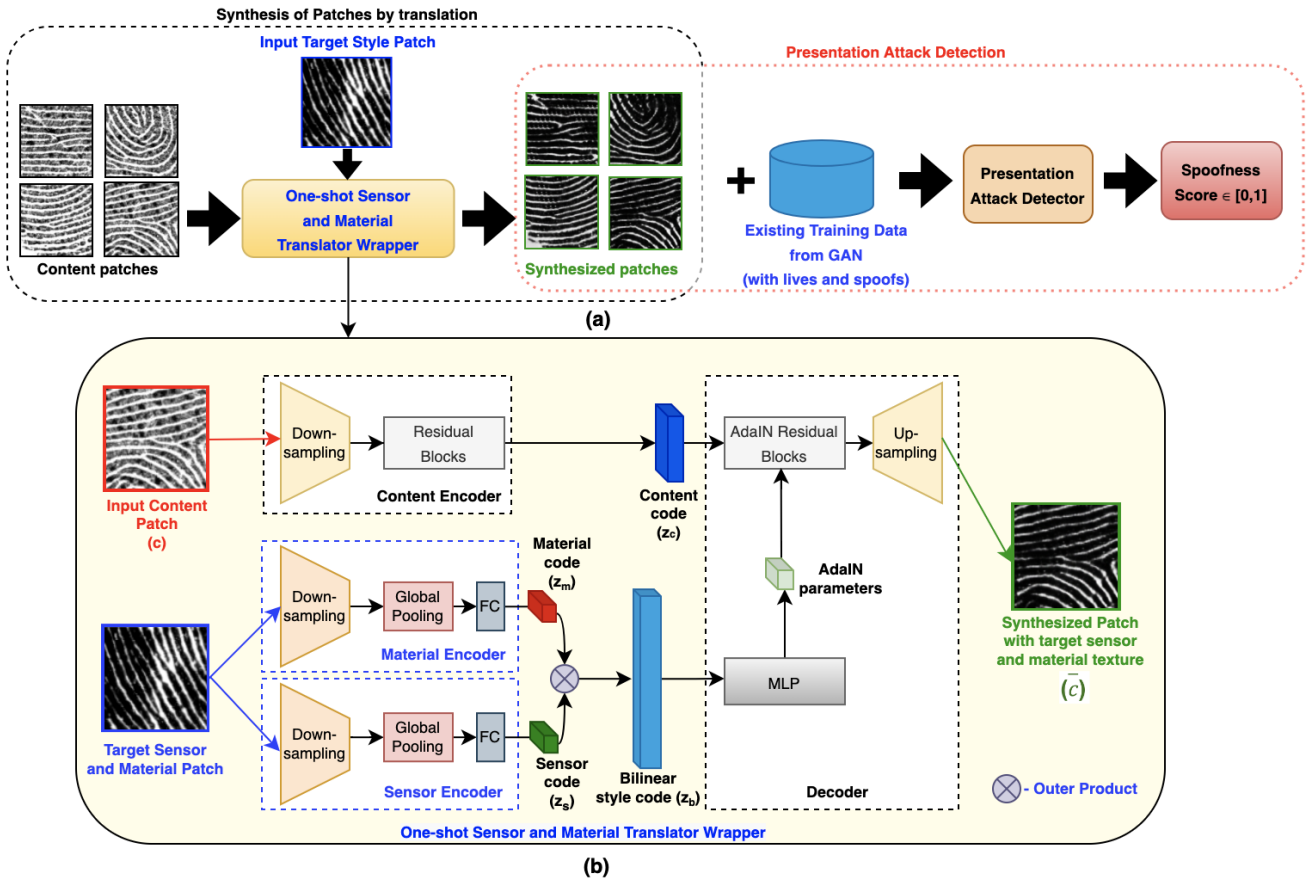


Figure 4.3: (a) An overview of the proposed pipeline with One-Shot Sensor and Material Translator (OSMT) wrapper over the PA detector for addition of synthesized patches belonging to the target sensor and material (b) Architecture of the Generator of One-Shot Sensor and Material Translator (OSMT) wrapper. A content patch $\mathbf{c}$ and a style patch (of the target sensor $\mathbf{s}$ and material $\mathbf{m}$) are passed into the content, sensor and material encoders respectively. The generated sensor and material codes ($\mathbf{z}_s$ and $\mathbf{z}_m$) are fused using a matrix outer product to form the bilinear style code $\mathbf{z}_b$. The content code $\mathbf{z}_c$ is processed through the decoder with AdaIN parameters from the style code $\mathbf{z}_b$ to generate the final output image $\overline{\mathbf{c}}$. $\overline{\mathbf{c}}$ contains the content of $\mathbf{c}$ and texture of sensor $\mathbf{s}$ and material $\mathbf{m}$.

disentanglement in the style feature space. Both sensor (E_s) and material (E_m) encoders have the same network. Their network consists of several strided convolutional layers for downsampling, followed by a global average pooling layer and a fully connected (FC) layer. The encoders E_s and E_m map the style patches to intermediate latent vectors $\{\mathbf{s}_1, \ldots, \mathbf{s}_{K1}\}$ and $\{\mathbf{m}_1, \ldots, \mathbf{m}_{K2}\}$, which are then converted to 8-dimensional sensor ($\mathbf{z}_s$) and material ($\mathbf{z}_m$) codes by computing their mean.

- **Bilinear Layer:** We combine the sensor ($\mathbf{z}_s$) and material ($\mathbf{z}_m$) codes using a bilinear layer B similar to [36] for computing the final bilinear style code $\mathbf{z}_b$. In the bilinear layer, the 8-dimensional codes $\mathbf{z}_s$ and $\mathbf{z}_m$ are combined using an outer product to produce an 8×8 intermediate output. This output is flattened using the vec operator as shown in Equation 4.2 to produce a 64-dimensional style code $\mathbf{z}_b$.

$$\mathbf{z}_b = B(\mathbf{z}_s, \mathbf{z}_m) = vec(\mathbf{z}_s \otimes \mathbf{z}_m) = vec(\mathbf{z}_s^T \mathbf{z}_m) \tag{4.2}$$

  The bilinear combination merges the sensor and material information but also captures the pairwise interactions between them.

- **Decoder:** The decoder F_c takes in both the content and style codes to produce the output image. The decoder consists of multiple adaptive instance normalization (AdaIN) residual blocks [27], followed by upsampling and convolutional layers. The AdaIN residual blocks use AdaIN [26] as the normalization layer. The affine parameters of AdaIN in Equation 4.3, are computed from the style code $\mathbf{z}_b$ using the multilayer perceptron network (MLP).

$$\mathrm{AdaIN}(z, \gamma, \beta) = \gamma \left(\frac{z - \mu(z)}{\sigma(z)} \right) + \beta \tag{4.3}$$

  where z is the activation from the previous convolutional layer, μ and σ are channelwise mean and standard deviation, γ and β are the affine parameters computed by the MLP.

  The computed μ and σ are used to perform affine transformations in each of the AdaIN residual blocks at the normalization layer. In this way, we infuse the style into the decoded content feature maps using the affine transformations in the AdaIN residual blocks to generate the final stylized fingerprint patch.

  As shown in Figure 4.3, Equation 4.1 now becomes

$$\overline{\mathbf{c}} = F_c\left(\mathbf{z}_c, \mathbf{z}_b\right) = F_c\left(E_c(\mathbf{c}), B(\mathbf{z}_s, \mathbf{z}_m)\right) \tag{4.4}$$

- **Sensor and Material Discriminators:** We utilize two multi-task adversarial discriminators [38] D_s and D_m for sensor and material respectively. This type of discriminator solves various binary classification tasks simultaneously. We determine whether the input is an original image from the known source class or a translation output from G in each task.

4.3.2.2 Formulation

We train the proposed OSMT wrapper by solving the following minimax optimization function:

$$\min_{D} \max_{G} \mathcal{L}_{\mathrm{GAN}}(D_s, D_m, G) + \lambda_{\mathrm{C}}\mathcal{L}_{\mathrm{C}}(G) + \lambda_{\mathrm{F}}\mathcal{L}_{\mathrm{F}}(G) + \lambda_{\mathrm{S}}\mathcal{L}_{\mathrm{sensor}}(G) + \lambda_{\mathrm{M}}\mathcal{L}_{\mathrm{material}}(G) \tag{4.5}$$

where $\mathcal{L}_{\mathrm{GAN}}$, $\mathcal{L}_{\mathrm{C}}$, $\mathcal{L}_{\mathrm{F}}$, $\mathcal{L}_{\mathrm{sensor}}$ and $\mathcal{L}_{\mathrm{material}}$ are the GAN loss, the content image reconstruction loss, the feature matching loss, the sensor and material contrastive loss respectively.

We use a GAN loss conditioned on material and sensor as given in Eqn. 4.6 . The superscripts of D - c_{ss}, c_{sm}, c_s and c_m denote the source sensor, source material, target sensor and target material.

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}(D_s, D_m, G) =& 0.5 * (E_{\mathbf{c}}\left[-\log D_s^{c_{ss}}(\mathbf{c})\right] + E_{\mathbf{c},\mathbf{s}}\left[\log\left(1 - D_s^{c_s}(\overline{\mathbf{c}})\right)\right] + \\ & 0.5 * (E_{\mathbf{c}}\left[-\log D_m^{c_{sm}}(\mathbf{c})\right] + E_{\mathbf{c},\mathbf{m}}\left[\log\left(1 - D_m^{c_m}(\overline{\mathbf{c}})\right)\right]\end{aligned} \tag{4.6}$$

We use a contrastive loss [11] on the sensor and material codes produced by the sensor and material encoders. It brings the embeddings belonging to the same material or sensor closer in the feature space and pushes them away if the classes are dissimilar, using the objective below:

$$\mathcal{L}(x_1, x_2) = (1 - y)\frac{1}{2}\left(D_w\right)^2 + (y)\frac{1}{2}\left\{\max\left(0, m - D_w\right)\right\}^2 \tag{4.7}$$

where x_1 and x_2 are the input images. When x_1 and x_2 belong to the same class $y = 1$, else $y = 0$. D_w is the euclidean distance between the embeddings and m is the margin.

The content reconstruction loss L_C ensures the content (fingerprint ridges) is retained in the synthesized image after translation. We pass the same input content image through all the three encoders - content, sensor and material. Thereby, forcing G to produce an output image identical to the input. We use L1 norm since it generates sharper images.

$$\mathcal{L}_{\mathrm{C}}(G) = E_{\mathbf{c}}\left[\|\mathbf{c} - G(\mathbf{c}, \{\mathbf{c}, \mathbf{c}\})\|_1^1\right] \tag{4.8}$$

In order to regularize the training, we use a feature matching loss for both sensor and material. The feature extractors D_{fs} and D_{fm} are constructed by removing the final prediction layer. We use D_{fs} and D_{fm} for extracting features from translated output $\overline{\mathbf{c}}$, target sensor $\{\mathbf{s}_1, \ldots, \mathbf{s}_{K1}\}$ and material images $\{\mathbf{m}_1, \ldots, \mathbf{m}_{K2}\}$ to minimize the loss below:

$$\mathcal{L}_{\mathrm{F}}(G) = E_{\mathbf{c},\mathbf{s}}[\|D_{fs}(\overline{\mathbf{c}})) - \sum_{K1}\frac{D_{fs}\left(\mathbf{s}_k\right)}{K1}\|_1^1] + E_{\mathbf{c},\mathbf{m}}[\|D_{fm}(\overline{\mathbf{c}}) - \sum_{K2}\frac{D_{fm}\left(\mathbf{m}_k\right)}{K2}\|_1^1] \tag{4.9}$$

4.4 Datasets

We utilize the LivDet (2015 [47], 2017 [48], 2019 [50] and 2021 [7]) fingerprint liveness datasets for our experiments. Some of these datasets contain additional spoof materials in the test set other than the ones in the train set. We only utilize the spoof materials common to both train and test data for our experiments.

For LivDet17, 19 and 21 datasets, we cannot perform 'cross-sensor' or 'cross-sensor and cross-material' experiments since there are only two optical sensors. For the above two experiments, we must remove the target sensor from the train data, leaving only one sensor to learn the sensor contrastive loss. Similarly, LivDet21 has only two materials in train data, so we cannot perform cross-material experiments.

LivDet17, 19 and 21 datasets also have disjoint spoof materials in the train-test split. Therefore, while testing, we should use the train data of the target sensor and material left out during training, as the test set. This test data has the same identities (content) as the training data across other sensors, which might not be a good evaluation method. Due to the above reasons, we performed 'cross-sensor and cross-material' experiments on the more suitable LivDet15 dataset, having more sensors and a train-test split with common materials and sensors. However, we can obtain the cross-material performances for LivDet17 and 19 as presented in section 5.4.

We also utilize the LivDet 2021 [7] fingerprint dataset for evaluating performance across fabrication techniques. LivDet 2021 consists of fingerprints captured through both Consensual and ScreenSpoof techniques. In the traditional consensual method, the mold is fabricated with user collaboration. The ScreenSpoof [8] technique uses a semi-consensual method to model realistic attacks and helps in better performance assessment. The steps for the ScreenSpoof capture are as follows:

- A snapshot of a latent fingerprint left on a smartphone screen is captured.
- After appropriate preprocessing (segmenting and enhancing), its negative image is printed on a transparent sheet for the mold, similar to the non-consensual process.

Pre-processing

We first resize the 1000 dpi images of Biometrika sensor in LivDet 2015 to 500 dpi.

We preprocess these images by segmenting the fingerprint from the background similar to [18] and then extract 150x150 minutiae patches. These minutiae patches are aligned and center cropped to 96x96 patches.

4.5 Experiments, Results and Analysis

We evaluate the performance of our proposed OSMT wrapper by utilizing a presentation attack classifier. We conduct experiments in three different settings - cross-material, cross-sensor and both cross-sensor and cross-material. With the help of these results, we demonstrate that the addition of synthetic fingerprint patches improves the sensor and material generalization of a PA detector.

4.5.1 Experimental Procedure

We utilize a leave-one-out strategy similar to [18] to evaluate our model's performance. Depending on the cross-experiment type, we leave the particular testing material, sensor, or both out of the train

data to simulate the real-world scenario. The left-out train data is then augmented to the test data of the particular material or sensor to increase the test size for the PA classifier. We train the OSMT wrapper with the rest of the data.

At inference of OSMT, we pass content patches and 50 minutiae patches[2] from a single target fingerprint to synthesize data of the target sensor and material. We choose content patches from the bonafide patches of the fixed sensor for cross-material experiments. For cross-sensor experiments, we select content patches from a sensor other than the test sensor. The PA detector is trained with and without these synthesized patches to verify the improvement.

4.5.2 Major Results

4.5.2.1 Comparison with State-of-the-Art

Table 4.1: Comparison of OSMT and SOTA cross-material performance (TDR (%) @ FDR = 0.1%) using EcoFlex, Body Double, and Play Doh spoof materials, without and with using synthesized spoof patches in the train set.

Training Set	**Testing Set**	**TDR (%) @ FDR = 0.1%**
Bonafide vs [EcoFlex + BD]	Bonafide vs PD	92.73
Bonafide vs [EcoFlex + BD + 50 Spoof PD Patches* + 35K UMT Synthesized PD Patches]	Bonafide vs PD	93.74
Bonafide vs [EcoFlex + BD + 50 Spoof PD Patches* + **35K OSMT Synthesized PD Patches**]	Bonafide vs PD	**94.7**
Bonafide vs [PD + EcoFlex]	Bonafide vs BD	81.78
Bonafide vs [PD + EcoFlex + 50 Spoof BD Patches* + 35K UMT Synthesized BD Patches]	Bonafide vs BD	**81.88**
Bonafide vs [PD + EcoFlex + 50 Spoof BD Patches* + **35K OSMT Synthesized BD Patches**]	Bonafide vs BD	80.04
Bonafide vs [PD + BD]	Bonafide vs EcoFlex	90.02
Bonafide vs [PD + BD + 50 Spoof EcoFlex Patches* + 35K UMT Synthesized EcoFlex Patches]	Bonafide vs EcoFlex	91.19
Bonafide vs [PD + BD + 50 Spoof EcoFlex Patches* + **35K OSMT Synthesized EcoFlex Patches**]	Bonafide vs EcoFlex	**93.9**

*50 Spoof EcoFlex / Body Double / Play Doh minutiae patches are generated from only 1 Spoof EcoFlex / Body Double / Play Doh image, respectively. PD = Play Doh, BD = Body Double

Our method utilizes 50 minutiae patches from a single fingerprint with a target material within the CrossMatch sensor for synthesis. We added 35000 synthesized patches to the train data for each new material.

[2]Our wrapper can utilize only one target patch. However, we use multiple patches to capture the variability in style.

We re-implemented UMT with minutiae patches for the baseline and observed that our UMT performance is better than the original implementation (with randomly cropped patches).

We suitably compare our cross-material performance with UMT [18] as the state-of-the-art model since we cannot compare our model with other SOTA methods - UMG [13], UMG+ARL [23]. UMG tries to achieve a different goal by synthesizing images through interpolation between two materials. However, we try to obtain results by extrapolating to materials and sensors outside the convex hull. ARL is completely different from our method, it could also be applied on top of OSMT.

Table 1. presents the improvement in TDR @ FDR = 0.1% on the addition of synthesized patches and compares our performance with the existing UMT wrapper. Here, we do not use the entire train data (across multiple sensors) of the wrapper but only use the train set of the CrossMatch sensor, excluding the target material, for suitable comparison to UMT.

4.5.2.2 Cross-Sensor Performance

We observe a consistent performance improvement across all sensors, and achieve an average improvement of 4.31% (TDR (%) @ FDR = 0.1%). Table 4.2 shows the cross-sensor performance without and with the synthesized patches of the target sensor.

While training, we include the synthesized data of the target sensor and the real data of the three sensors, excluding the target sensor. In the synthesized dataset of the test sensor, we include both the synthesized live and synthesized spoof material patches. We supply 50 patches each, for live and every spoof material belonging to the target sensor to the wrapper for synthesis. The number of synthesized patches of the target sensor equals the number of actual live and spoof patches within the sensor in the LivDet2015 dataset.

Table 4.2: Cross-sensor performance (TDR (%) @ FDR = 0.1%) without and with synthesized patches of the corresponding test sensor in the train set

Sensor	**TDR (%) @ FDR = 0.1%**		
	Digital Persona	**Green Bit**	**Bio-metrika**
Without synthesized	6.66	67.59	35.95
With UMT synthesized	8.07	57	37.88
With OSMT synthesized	9.05	75.86	38.22

4.5.2.3 Cross-Material Performance

We evaluate the cross-material performance of LivDet 2017 and 2019 datasets similar to section 4.5.2.1. For each material, we utilize 35000 and 25000 synthesized patches generated using live patches from GreenBit and Digital Persona sensors respectively.

Tables 4.3 and 4.4 presents the results for cross-material performance on LivDet2017 and LivDet2019 respectively.

Table 4.3: Cross-material performance (TDR@FDR=0.1%) without and with synthesized patches for LivDet 2017

Material	**TDR (%) @ FDR = 0.1%**	
	Without synthesized	**With OSMT synthesized**
Body Double	86.58	85.51
Ecoflex	56.41	46.91
Wood Glue	75.94	76.76

Table 4.4: Cross-material performance (TDR@FDR = 0.1%) without and with synthesized patches for LivDet 2019

Material	**TDR (%) @ FDR = 0.1%**	
	Without synthesized	**With OSMT synthesized**
Body Double	99.48	99.96
Ecoflex	71.41	66.25
Wood Glue	20.52	22.67
Ecoflex-0050	44.37	45.34
Gelatine	48	55.83
Latex	49.2	62.17

4.5.2.4 Cross-Sensor and Cross-Material Performance

While evaluating cross-sensor and cross-material performance, we remove both the lives and spoof data of the unknown sensor from the train set. The data belonging to the novel material within all the other sensors in train data is also removed. The number of synthesized patches is approximately equal to the number of patches in each spoof material for the given sensor.

To generate patches of unknown sensor and unknown material, we supply two sets of 50 minutiae patches each from (i) unknown material and known sensor and (ii) known material and unknown sensor. We utilize 15,000 synthesized patches for every unknown sensor-material combination. CrossMatch sensor in Table 4.5 is excluded due to the non-overlap of materials in the dataset.

Figure 4.4 presents synthesized patches generated by OSMT using cross-sensor and cross-material experiments. By splitting and combining sensor and material codes, our method can generate various combinations that UMT cannot.

Table 4.5: Cross-sensor and cross-material performance (TDR (%) @ FDR = 0.1%) without and with OSMT synthesized patches of the corresponding test material and sensor in training

Materials	GreenBit		Digital Persona		Biometrika	
	Without synthesized	**With synthesized OSMT**	**Without synthesized**	**With synthesized OSMT**	**Without synthesized**	**With synthesized OSMT**
Ecoflex 00-50	96.07	97.7	10.46	7.06	47.34	58.6
Gelatine	81.52	84.19	4.37	18.53	21.74	24.27
WoodGlue	61.32	51.15	3.81	2.06	6.44	5.91
Latex	91.51	91.6	3.81	9.81	69.07	80.8

Comparison with Classical Methods Baseline

Table 4.6: Cross-sensor and cross-material performance (TDR (%) @ FDR = 0.1%) without and with synthesized patches of the corresponding test material and sensor in training and comparison with their corresponding classical linear method baseline

Materials	GreenBit		Digital Persona		Biometrika	
	With linear synthesized	**With synthesized OSMT**	**With linear synthesized**	**With synthesized OSMT**	**With linear synthesized**	**With synthesized OSMT**
Ecoflex 00-50	75.67	97.7	6.57	7.06	51.51	58.6
Gelatine	81.82	84.19	2.34	18.53	20.73	24.27
WoodGlue	47.12	51.15	0.37	2.06	3.9	5.91
Latex	65.74	91.6	2.13	9.81	58.69	80.8

4.5.3 Ablation Studies

4.5.3.1 Varying Number of Synthesized Patches Used

We study the variation in cross-sensor and cross-material performance by varying the number of synthesized patches added while training for the combination - GreenBit and Gelatine. We observe a performance improvement till 15k patches and then a decrement with the addition of 30k patches, as shown in Table 4.7.

We conclude that excessive synthetic patches could overwork the information already present, and the extra noise would affect the overall performance.

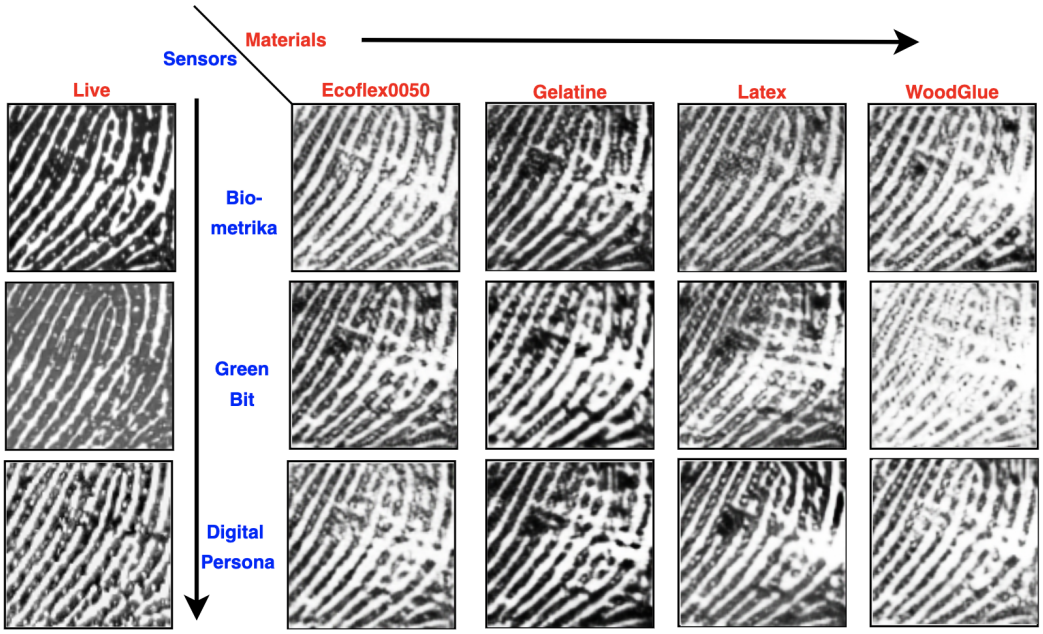


Figure 4.4: Synthesized patches for cross-sensor and cross-material combinations using our OSMT wrapper. Bonafide images for the corresponding sensors are in the first column.

Table 4.7: Variation in cross-sensor and cross-material performance (TDR (%) @ FDR = 0.1%) for GreenBit-Gelatine combination with 5k, 15k and 30k patches

Without synthesized	With 5k	With 15k	With 30k
81.52	82.29	84.19	75.39

4.5.3.2 Performance across Fabrication Techniques

We utilize LivDet 2015 and LivDet 2021 [7] datasets for training and testing respectively, to evaluate the performance across different fabrication techniques. To generate cross-sensor and cross-material synthesized patches, we use patches from GreenBit-Live and CrossMatch-Body Double of LivDet 2015, to generate GreenBit-Body Double patches. We use GreenBit-Body Double patches from the test set of LivDet 2021 for evaluating the cross-sensor and cross-material performance.

LivDet 2021 consists of fingerprints captured through both Consensual and ScreenSpoof techniques. The ScreenSpoof [8] technique models realistic attacks and helps in better performance assessment. We supply target patches captured through the Consensual method only, due to the non-availability of ScreenSpoof data in LivDet 2015. However, we evaluate our approach on LivDet 2021 fingerprints captured using both techniques, as demonstrated in Table 4.8.

Table 4.8: Performance across fabrication techniques for training on LivDet 2015 and testing on LivDet 2021 dataset. Cross-sensor and cross-material performance (TDR (%) @ FDR = 0.1%) without and with synthesized patches for two methods of capture - Consensual and ScreenSpoof.

Sensor-Material	Without synthesized		With synthesized	
	Consensual	Screen Spoof	Consensual	Screen Spoof
GreenBit-BodyDouble	49.93	59.74	62.07	65.18

4.5.4 Qualitative Results

4.5.4.1 Incorrectly and Correctly Classified Patches by OSMT

In Figure 4.5, we present a few correctly predicted and misclassified fingerprint samples. We observe that PA fingerprints with a porous and lighter texture get misclassified as live. Fake fingerprints with missing ridges and artifacts are easily detected by our method.

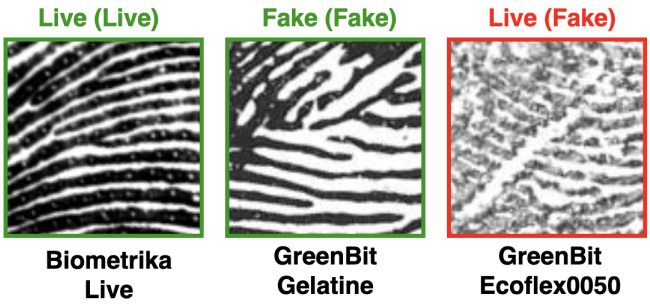


Figure 4.5: Samples of correctly classified and misclassified predictions of our classifier on live and PA fingerprint patches with the title - predicted label (true label).

4.5.4.2 Classification Results of UMT vs. OSMT

We present a few example images that are correctly classified by OSMT, but fails to be classified by UMT.

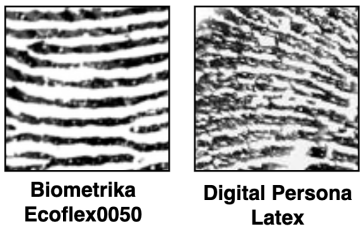


Figure 4.6: Correctly classified samples by OSMT but misclassified by UMT.

We observe that images similar to live are being misclassified by UMT, but not by our proposed method.

4.5.5 Exceptions

We observe that the classification accuracy for the Digital Persona sensor is much lower compared to other sensors. This is mainly due to its small-sized sensor used to capture the fingerprint, thereby producing smaller images. We also observe that materials like WoodGlue and Ecoflex do not have very good accuracy improvement. These materials produce transparent spoofs allowing the live fingerprint color to pass through, making them harder to distinguish from live fingerprints.

4.6 Implementation Details

The discriminators D_s and D_m consists of an output dimension equal to the number of sensors and materials in the training dataset. We set $\lambda_{\mathrm{S}} = 1$ and $\lambda_{\mathrm{M}} = 1$ and batch size as 24 for OSMT. Other hyperparameters are set identical to [38].

We use a MobileNetV2 classifier pretrained on ImageNet dataset [14] and remove the last layer and replace it with two output neurons for live and PA classes. The classifier trained with an Adam optimizer with a learning rate of $1\mathrm{e}{-3}$ and a batch size of 200 while training across multiple sensors. For cross-material experiments, we use a train and test batch size of 64 and 120 respectively.

4.6.1 Network Details

We utilize the framework similar to FUNIT [38] as mentioned in Section 4. However, we use the network layers of MUNIT [27] for our generator. MUNIT contains fewer layers in the generator which is suitable for us due to the lower complexity required for learning fingerprints.

4.7 Summary

The progressive demand for automatic fingerprint recognition systems has increased the variety of presentation attacks. Earlier FPAD solutions utilize style transfer wrappers over PA detectors by transferring the unknown style to content fingerprint patches.

We model presentation attacks as a combination of two underlying components - material and sensor, rather than the entire style. By utilizing the disentanglement approach of our OSMT wrapper, we can generate synthetic patches on introducing a new sensor, material or both.

We reduce the number of style representations to be learned from **nxm** to **n+m**. In this way, OSMT can generate a tremendous amount of data with less samples. We also observe our method improves cross-sensor and cross-material generalization.

Chapter 5

Conclusions and Future Work

Fingerprint presentation attack detectors are usually trained on datasets containing limited live and spoof materials. However, attackers can bypass these systems using unknown spoof materials or attack methods other than those encountered during training. Such attacks can be detected by improving the generalization capability of a fingerprint presentation attack detection system.

In this thesis, we address the problem of cross-sensor and cross-material FPAD generalization using a synthesis-based approach. We improve presentation attack detection by the addition of synthesized patches of the target sensor and material while training the PAD classifier. Existing approaches learn the whole fingerprint style, however, we utilize a decomposition-based approach. Thus helping us learn the corresponding sensor and material representations. By utilizing this approach, we can generate synthetic patches upon introducing a new sensor, material or both.

We demonstrate two different methods of fingerprint factorization - traditional and deep-learning based, in this thesis.

In chapter 3, we discuss traditional methods for factorization of fingerprint into sensor and material representations using tensor decomposition called *tensorprints*. The factorized sensor and material codes are used to synthesis new images by translation. This establishes a baseline using machine learning for our hypothesis.

In chapter 4, we propose a deep-learning based augmentation wrapper for the disentanglement of fingerprint style called OSMT and present its superiority over the traditional factorization techniques. The OSMT (One-shot Sensor and Material Translator) wrapper improves cross-sensor and cross-material PAD, utilizing one fingerprint image of the target sensor and material.

Our approach aims to improve performance by *extrapolating* to sensors and materials outside the convex hull. It also reduces computational complexity by generating compact representations and utilizing lesser combinations of sensors and materials to produce several styles. This allows us to explore a much larger and diverse feature space, and generates a large variety of samples using the same limited amount of data, which helps improve generalization further.

In the future, we would like to incorporate open-set detection as a part of the presentation attack detection pipeline. This would allow us to flag unknown sensors and materials and later pass them to

our wrapper. This would reduce the computational complexity of the entire system, as synthesis using OSMT and retraining the classifier will be required only on detecting an unknown sensor or material.

Related Publications

- One-Shot Sensor and Material Translator : A Bilinear Decomposer for Fingerprint Presentation Attack Generalization
  Gowri Lekshmy, Anoop Namboodiri, *International Joint Conference on Biometrics (IJCB)*, 10-13 October 2022, Abu Dhabi, United Arab Emirates

Bibliography

[1] International standards organization, "iso/iec 30107- 1:2016, information technology—biometric presentation attack detection—part 1: Framework", 2016.

[2] A. Abhyankar and S. Schuckers. Integrating a wavelet based perspiration liveness check with fingerprint recognition. *Pattern Recognition*, 42(3):452–464, 2009.

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

[4] S. S. Arora, A. K. Jain, and N. G. Paulter. Gold fingers: 3d targets for evaluating capacitive readers. *IEEE Transactions on Information Forensics and Security*, 12(9):2067–2077, 2017.

[5] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni. Fake fingerprint detection by odor analysis. In *ICB*, 2006.

[6] K. Cao and A. K. Jain. Hacking mobile phones using 2 d printed fingerprints. 2016.

[7] R. Casula, M. Micheletto, G. Orrù, R. Delussu, S. Concas, A. Panzino, and G. Marcialis. Livdet 2021 fingerprint liveness detection competition – into the unknown, 08 2021.

[8] R. Casula, G. Orrù, D. Angioni, X. Feng, G. L. Marcialis, and F. Roli. Are spoofs from latent fingerprints a real threat for the best state-of-art liveness detectors? *CoRR*, abs/2007.03397, 2020.

[9] Y. Chen, A. Jain, and S. Dass. Fingerprint deformation for spoof detection. In *Biometric symposium*, volume 21, 2005.

[10] Y. Cheng and K. V. Larin. Artificial fingerprint recognition by using optical coherence tomography with autocorrelation analysis. *Appl. Opt.*, 45(36):9238–9245, Dec 2006.

[11] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005.

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

[13] T. Chugh and A. K. Jain. Fingerprint spoof detector generalization. *IEEE Transactions on Information Forensics and Security*, 16:42–55, 2021.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[15] Y. Ding and A. Ross. An ensemble of one-class svms for fingerprint spoof detection across different fabrication materials. pages 1–6, 12 2016.

[16] J. J. Engelsma, K. Cao, and A. K. Jain. Raspireader: Open source fingerprint reader. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2511–2524, 2019.

[17] W. Freeman and J. Tenenbaum. Learning bilinear models for two-factor problems in vision. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 554–560, 1997.

[18] R. Gajawada, A. Popli, T. Chugh, A. Namboodiri, and A. K. Jain. Universal material translator: Towards spoof fingerprint generalization. In *2019 International Conference on Biometrics (ICB)*, pages 1–8, 2019.

[19] L. Gatys, A. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv*, 08 2015.

[20] L. Ghiani, A. Hadid, G. L. Marcialis, and F. Roli. Fingerprint liveness detection using binarized statistical image features. In *2013 IEEE sixth international conference on biometrics: theory, applications and systems (BTAS)*, pages 1–6. IEEE, 2013.

[21] L. Ghiani, G. L. Marcialis, and F. Roli. Fingerprint liveness detection by local phase quantization. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 537–540, 2012.

[22] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva. Fingerprint liveness detection based on weber local image descriptor. In *2013 IEEE workshop on biometric measurements and systems for security and medical applications*, pages 46–50. IEEE, 2013.

[23] S. A. Grosz, T. Chugh, and A. K. Jain. Fingerprint presentation attack detection: A sensor and material agnostic approach. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10, 2020.

[24] K. Hayashi, T. Takenouchi, T. Shibata, Y. Kamiya, D. Kato, K. Kunieda, K. Yamada, and K. Ikeda. Exponential family tensor factorization: An online extension and applications. *Knowledge and Information Systems*, 33, 10 2012.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[26] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[27] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[28] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing.

[29] J. Kannala and E. Rahtu. Bsif: Binarized statistical image features. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 1363–1366. IEEE, 2012.

[30] J. Kossaifi, Y. Panagakis, and M. Pantic. Tensorly: Tensor learning in python. *CoRR*, abs/1610.09555, 2016.

[31] D. Kotovenko, A. Sanakoyeu, S. Lang, and B. Ommer. Content and style disentanglement for artistic style transfer. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4421–4430, 2019.

[32] P. Krishnan. *Learning Representations for Word Images*. PhD thesis, International Institute of Information Technology, 2020.

[33] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[34] P. D. Lapsley, J. A. Lee, D. F. Pare, and J. N. Hoffman. Anti-fraud biometric scanner that accurately detects blood flow. *US Patent, 5,737,439*, 1998.

[35] Y. Li, Y. Du, and X. Lin. Kernel-based multifactor analysis for image synthesis and recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 114–119 Vol. 1, 2005.

[36] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1457, 2015.

[37] M. Liu, T. M. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *CoRR*, abs/1703.00848, 2017.

[38] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz. Few-shot unsupervised image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[39] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. Handbook of fingerprint recognition. In *Springer Professional Computing*, 2003.

[40] E. Marasco and A. Ross. A survey on antispoofing schemes for fingerprint recognition systems. *ACM Comput. Surv.*, 47(2), nov 2014.

[41] E. Marasco and C. Sansone. On the robustness of fingerprint liveness detection algorithms against new materials used for spoofing. In *BIOSIGNALS*, 2011.

[42] E. Marasco and C. Sansone. Combining perspiration- and morphology-based static features for fingerprint liveness detection. *Pattern Recognition Letters*, 33(9):1148–1156, 2012.

[43] S. Marcel, M. S. Nixon, and S. Z. Li. *Handbook of Biometric Anti-Spoofing: Trusted Biometrics under Spoofing Attacks*. Springer Publishing Company, Incorporated, 2014.

[44] G. L. Marcialis, F. Roli, and A. Tidu. Analysis of fingerprint pores for vitality detection. *2010 20th International Conference on Pattern Recognition*, pages 1289–1292, 2010.

[45] M. Mathieu, J. J. Zhao, P. Sprechmann, A. Ramesh, and Y. LeCun. Disentangling factors of variation in deep representations using adversarial training. *CoRR*, abs/1611.03383, 2016.

[46] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino. Impact of artificial "gummy" fingers on fingerprint systems. In *IS&T/SPIE Electronic Imaging*, 2002.

[47] V. Mura, L. Ghiani, G. L. Marcialis, F. Roli, D. A. Yambay, and S. A. Schuckers. Livdet 2015 fingerprint liveness detection competition 2015. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–6, 2015.

[48] V. Mura, G. Orrù, R. Casula, A. Sibiriu, G. Loi, P. Tuveri, L. Ghiani, and G. Marcialis. Livdet 2017 fingerprint liveness detection competition 2017. pages 297–302, 02 2018.

[49] R. F. Nogueira, R. de Alencar Lotufo, and R. Campos Machado. Fingerprint liveness detection using convolutional neural networks. *Trans. Info. For. Sec.*, 11(6):1206–1213, jun 2016.

[50] G. Orrù, R. Casula, P. Tuveri, C. Bazzoni, G. Dessalvi, M. Micheletto, L. Ghiani, and G. Marcialis. Livdet in action - fingerprint liveness detection competition 2019. pages 1–6, 06 2019.

[51] F. Pala and B. Bhanu. *Deep Triplet Embedding Representations for Liveness Detection*, pages 287–307. Springer International Publishing, Cham, 2017.

[52] A. Rattani, W. J. Scheirer, and A. Ross. Open set fingerprint spoof detection across novel fabrication materials. *Trans. Info. For. Sec.*, 10(11):2447–2460, nov 2015.

[53] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[54] J. B. Tenenbaum and W. T. Freeman. Separating style and content. In *NIPS*, 1996.

[55] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[56] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

[57] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, 2016.

[58] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4105–4113, 2017.

[59] M. Vasilescu. Human motion signatures: analysis, synthesis, recognition. In *2002 International Conference on Pattern Recognition*, volume 3, pages 456–460 vol.3, 2002.

[60] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, page 447–460, Berlin, Heidelberg, 2002. Springer-Verlag.

[61] M. A. O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. 03 2002.

[62] X. Wu, H. Huang, V. M. Patel, R. He, and Z. Sun. Disentangled variational representation for heterogeneous face recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9005–9012, Jul. 2019.

[63] S. Yoon, J. Feng, and A. Jain. Altered fingerprints: Analysis and detection. *IEEE transactions on pattern analysis and machine intelligence*, 34:451–64, 07 2011.

[64] Y. Zhang, W. Cai, and Y. Zhang. Separating style and content for generalized style transfer. *CoRR*, abs/1711.06454, 2017.

[65] Y. Zhang, D. Shi, X. Zhan, D. Cao, K. Zhu, and Z. Li. Slim-rescnn: A deep residual convolutional neural network for fingerprint liveness detection. *IEEE Access*, PP:1–1, 07 2019.