8-2023

FedBiometric: Image Features Based Biometric Presentation Attack Detection Using Hybrid CNNs-SVM in Federated Learning

S M Sarwar
The University of Texas Rio Grande Valley

FedBiometric: Image Features Based Biometric Presentation Attack Detection Using Hybrid CNNs-SVM in Federated Learning

A Thesis

by

S M Sarwar

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: Computer Science

The University of Texas Rio Grande Valley

August 2023

FedBiometric: Image Features Based Biometric Presentation Attack Detection Using Hybrid CNNs-SVM in Federated Learning

A Thesis
by
S M Sarwar

COMMITTEE MEMBERS

Dr. Emmett Tomai

Chair of Committee

Dr. Bin Fu

Committee Member

Dr. Zhixiang Chen

Committee Member

August 2023

ABSTRACT

Sarwar, S M, FedBiometric: Image Features Based Biometric Presentation Attack Detection Using Hybrid CNNs-SVM in Federated Learning. Master of Science (MS), August, 2023, 87 pp., 10 tables, 39 figures, references, 198 titles.

In the past few years, biometric identification systems have become popular for personal, national, and global security. In addition to other biometric modalities, facial and fingerprint recognition have gained popularity due to their uniqueness, stability, convenience, and cost-effectiveness compared to other biometric modalities. However, the evolution of fake biometrics, such as printed materials, 2D or 3D faces, makeup, and cosmetics, has brought new challenges. As a result of these modifications, several facial and fingerprint Presentation Attack Detection methods have been proposed to distinguish between live and spoof faces or fingerprints. Federated learning can play a significant role in this problem due to its distributed learning setting and privacy-preserving advantages. This work proposes a hybrid ResNet50-SVM based federated learning model for facial Presentation Attack Detection utilizing Local Binary Pattern (LBP), or Gabor filter-based extracted image features. For fingerprint Presentation Attack Detection (PAD), this work proposes a hybrid CNN-SVM based federated learning model utilizing Local Binary Pattern (LBP), or Histograms of Oriented Gradient (HOG)-based extracted image features.

DEDICATION

At first, I would like to dedicate this thesis to my family, particularly my mother. With her support, I could successfully complete my master's degree. Next me, who is a first-generation student and has overcome all financial barriers, family problems, and mental problems while also sustaining myself alone in the USA to finish his graduation studies. My parents tried their best for me, and without their support, my journey would have been more difficult. In addition, I would like to dedicate this to the four family members I lost in the last two years but was unable to attend their funerals. These individuals include my father, grandfather, maternal uncle, and one additional relative.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Emmett Tomai, for his direction, feedback, and suggestions during my research work. Without Prof. Tomai, I could not have made this today. He was my last hope for any logistical supports. With his support, I explored different research conferences and programs. My thanks go to my thesis committee members, Dr. Bin Fu and Dr. Zhixiang Chen. Their advice, input, and comments on my dissertation helped to ensure the quality of my research work.

I would like to thank Dr. Honglu Jiang, who was my first supervisor at UTRGV and is now at Miami University. Also, I would like to thank Dr. Qi Lu for being my supervisor during a transitional period in the Fall of 2022.

I would like to thank Mr. Carlton Nelson, who is a Research and Instruction Librarian, UTRGV. Mr. Carl helped boost my searching skills and shared different research resources, which helped me finish this work. I would like to thank Ms. Lourdes Servantes, former associate director of the UTRGV Career Center, who helped me to build my network within research communities.

Sincerely, I acknowledge UTRGV's courtesy in providing me with a Presidential Research Fellowship (PRF) to do my research work, which just boosted my research skills to the next level. Without PRF support, it would be hard for me to finish my degree. I would like to thank Dr. Constatine M. Tarawneh (Associate Dean for Research and Graduate Programs), Dr. Emmett Tomai, and Dr. Honglu Jiang for this PRF support.

Also, I would like to thank the UTRGV Writing Center for reviewing my writeup and the UTRGV Career Center for helping me prepare research statements for different programs.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In the past few years, biometric identification systems have become popular for personal, national, and global security. Over recent years, biometric authentication systems have gained widespread acceptance for personal (mobile devices, access control systems), national (law enforcement, voter registration), and global security (visa applications, passport control). But the evolution of fake biometrics, such as printed materials, 2D or 3D faces, makeup, cosmetics, gelatine, silicon, woodglue, and latex, has brought new challenges. Day by day, introdurs are changing their spoofing style, and privacy is very important to train this kind of spoofing dataset. Nowadays, people are concerned about their privacy. So Federated Learning (FL) could bring a new solution to preserve users privacy because, in FL setting, users don't share their data; they only share updated parameters with the central server. In this thesis, we worked with two biometrics: facial and fingerprint.

1.1 Facial presentation attack detection

In addition to other biometric modalities, facial recognition has gained popularity due to its uniqueness, stability, convenience, and cost-effectiveness compared to other biometric modalities. However, the evolution of fake biometrics, such as printed materials, 2D or 3D faces, makeup, and cosmetics, has brought new challenges. As a result of these modifications, several facial Presentation Attack Detection methods have been proposed to distinguish between live and spoof faces. Federated Learning could play a significant role in this problem due to its distributed learning setting and privacy-preserving advantages. This work proposes a hybrid ResNet50-SVM based Federated Learning model for facial Presentation Attack Detection

(PAD) utilizing local binary pattern (LBP) or Gabor Filter based extracted image features.

1.2 Fingerprint Presentation Attack Detection

Fingerprints have become recognized as a popular biometric trait, alongside other traits such as the iris, face, retina, voice, signature, etc., because of their uniqueness, stability, convenience (touch or swipe), and cost-effectiveness compared to other biometric modalities. However, the evolution of fake biometrics such as gelatine, silicon, woodglue, and latex has brought new challenges. As a result of these modifications, several fingerprint Presentation Attack Detection methods have been proposed to distinguish between fake and spoof fingerprints. Federated Learning could play a significant role in this problem due to its collaborative learning method and privacy-preserving advantages. This work proposes a hybrid CNN-SVM based federated learning algorithm that uses Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) features of images.

1.3 Contributions in This Proposed Research

1.3.1 Facial Presentation Attack Detection

- To our best knowledge, this paper proposes ResNet50-SVM in federated learning setting for the first time.
- We evaluated the accuracy matrix of various CNN-SVM models like MobilenetV2-SVM and VGG16-SVM.
- We compared our work to other state-of-the-art methods and observed that, in distributed machine learning settings like FL, it had higher or similar validation accuracy.

1.3.2 Fingerprint Presentation Attack Detection

- To our best knowledge, this paper proposes CNN-SVM in federated learning setting on a fingerprint spoofing dataset for the first time.

- We evaluated the accuracy matrix of various CNN-SVM models like MobilenetV2-SVM and VGG16-SVM.
- We compared our work to other state-of-the-art methods and observed that, in distributed machine learning settings like FL, it had higher or similar validation accuracy.

CHAPTER II

BACKGROUND

This chapter presents an in-depth explanation of the contents, equipments, and methods used to prepare for this study. In order to proceed with the thesis, it is necessary to know these concepts.

2.1 Biometric Presentation Attacks

Biometric technologies, which use biological and behavioral traits to identify people, are widely used in security systems. Besides the well-known face and fingerprint biometrics, there are many more, including a person's DNA, voice, iris, palm, hand vein pattern, voice pattern signature, heart rate, gait, hand/finger geometry, keystroke pattern, signature and ear [1] [2]. Davis et al. [3] worked on creating a system that could automatically identify telephone-quality digits when a single person spoke them at normal speech speeds. In 1966, Bledsoe et al. [4] proposed a model for facial recognition from Panoramic Research Inc. After that, in 1971, Kelly, at Stanford University, came up with the idea of a computer system for identifying people [5]. As a result of these particular studies, the use of biometrics is now possible in a wide variety of contexts, including forensics, border and access control, surveillance, and online commerce.

Biometric systems appear more and more every year in different places like airports, laptops, and mobile phones. As a result, people are becoming more familiar with how these technologies work in everyday life, and as a result, their security weaknesses are becoming more widely recognized by everyone. Another reason is that over the internet, it is easy to find any tutorial or blog outlining how to make anything spoofable.

In July 2023, Statista [6] published an overall summary of the use of different types of biometric methods in application domains for the year of 2018. Fig 2.1 depicts the percentage of applications using different types of biometric methods in the USA and from this figure, it is

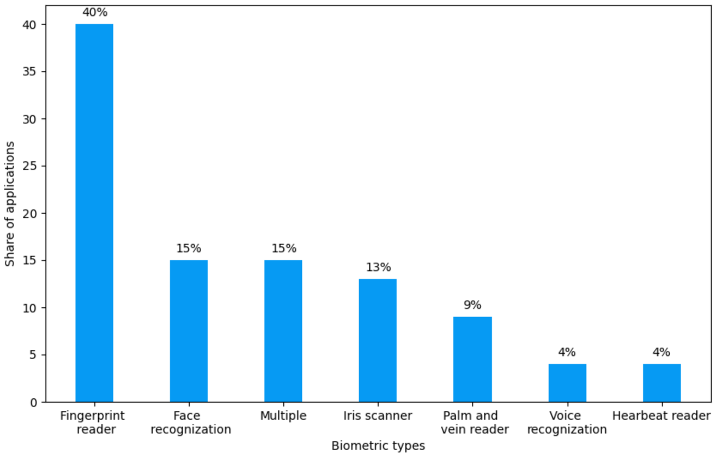


Figure 2.1: Percentage of applications using different types of biometric methods in the USA (2018).

easy to get that fingerprint and face biometrics are the most used. In India, bank customers have lost more than $700,000 as a result of fraud committed at the time of these transactions, and it happened due to the biometric authentication system. The statistics revealed by the State Bank of India, which had the most Aadhaar-based transactions during these five years of 2018–2023, also showed that it was also most exposed to these types of fraudulent activities [7].

DNA Spoofing is the next-generation threat to DNA privacy and genetic surveillance. At the Stranger Visions project, Dewey-Hagborg presented that it is possible to extract computer-generated 3-D portraits from genetic footprints [8]. Biometric Presentation Attacks (PAs or Spoof Attacks) target to interfere with the biometric system by presenting to the biometric capture subsystem.

2.2 Facial Presentation Attacks

Face spoofing is becoming very popular nowadays because it is easy to make a fake face using advanced techniques and because it is possible to manage spoofing materials at low costs for hackers. Face spoofing is classified into two categories: 2D spoofing (such as printed photos [9], [10] and live video [11]) and 3D spoofing [12] (such as latex, paper, and silicone). Video Attacks (sometimes called Replay Attacks) are a step up from simple photo spoofs. In Video Attacks, the attacker doesn't use a still photo but instead plays a live video of the real client

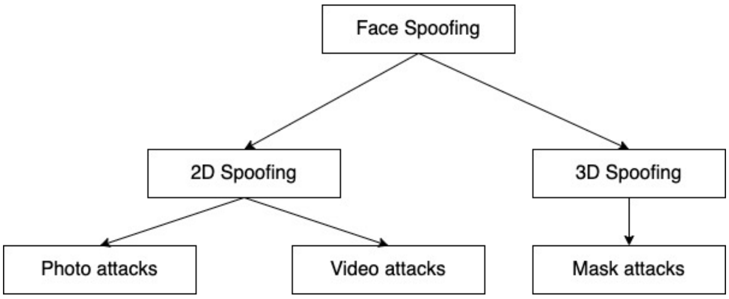


Figure 2.2: Classification of face spoofing techniques.

using a digital device (such as a mobile phone, tablet, or laptop). Fig. 2.2 indicates all of the types of face spoofing [13] [14].

According to the Daily Mail, one black person robbed betting shops using a latex mask and presenting himself as a white man [15]. Nguyen Minh Duc introduced a Bypass Model to test three laptops' cameras produced by Lenovo, Asus, and Toshiba security by using fake faces, and from their experiment, it is visible that users are at risk of securing their personal identity [16]. In May 2023, researchers found that 40% of all existing phones (including those from Honor, Motorola, Nokia, Oppo, Samsung, Vivo, and Xiaomi [17]) could be easily unlocked by a 2D printed facial image.

2.3 Facial Presentation Attacks Detection

Facial Presentation Attack Detection (PAD) refers to the process of identifying and distinguishing between live face and spoof or fake faces. Facial PAD utilize materials such as printed photos (2D images), masks (3D images) [18], makeup, and latex. Sometimes hackers use live video [19–21] to spoof authentication systems. Facial PAD demonstrates notable effectiveness when using deep learning-based approaches such as Convolutional Neural Networks (CNNs) [22], texture analysis, liveness detection, 3D face analysis [18], motion analysis, and spectrum analysis. Fig. 2.3 presents different categories of state-of-the-art face PAD techniques [23].

Hardware-Based PAD Techniques: Sensor characteristics, blink detection, and challenge response are different types of hardware-based PAD techniques. Raghavendra et al. [24] pre-

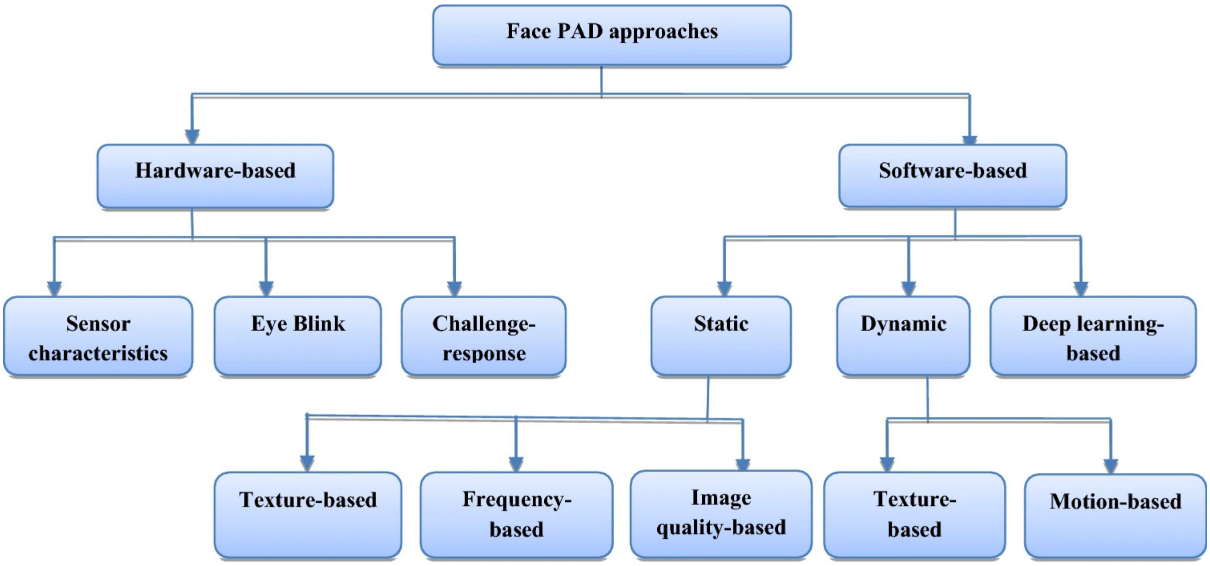


Figure 2.3: Classification of state-of-the-art face PAD techniques.

sented a light field camera (LFC) to distinguish presentation attacks (or spoof attacks). They used how the LFC changes the focus between different levels (or focuses) of a face image, which can be used to recognize fake images. Gang et al. [25] proposed a Conditional Random Fields (CRFs)-based approach to detect blinking activity and used a web camera for capturing video clips. Blicking activity could be classified as open, half-open, or c losed. To distinguish between eye states, they used a linear chain structure of CRFs. Lagorio et al. [26] presented a 3D face sensor-based liveness detection model where they measured the first-order statistics of the 3D surface curvature for a facial image, and the proposed model outperformed for both 2D and 3D facial images. Hardware-based techniques have some drawbacks, such as being ineffective for video attacks, having a high computation cost, and needing dedicated hardware.

Software-Based PAD Techniques: Software-based methods have low costs for computation, are less responsive to face regions, and work for both photo and video attacks. Waris et al. [27] came up with a rotation-invariant uniform LBP (neighboring pixels, P = 16; radius, R = 2) with a Gabor filter and Gray Level Co-occurrence Matrix (GLCM)-based approach for video or reply attack, and the proposed models performed better than other existing state-of-the-art methods on the REPLY-ATTACK dataset. Multi-scale local binary patterns (LBP) were used for encoding the micro-texture patterns into an enhanced feature histogram, which was then fed into Support Vector Machine (SVM) to classify fake or live images and achieved 98.0%

accuracy to detect fake images [28]. Zhang et al. [29] used multiple Difference of Gaussian (DoG) filters to get off the high-frequency information from facial images, which was then fed into SVM and tested with diverse attacks such as photo attacks (wraped and cut photos) and video attacks.

In September 2021, Apple updated the Face ID anti-spoofing model for its iPhone and iPad devices to better protect users against 3D face spoofing attacks [30]. Google introduced improved face anti-spoofing technology to secure their users more robustly on their Pixel phones [31].

2.4 Fingerprint Presentation Attacks

Among all biometric identifiers, fingerprints are the most widely used to identify people. Today, people personally use fingerprints to unlock their mobile phones, laptops, attendance systems, and many more. Nation-level uses include criminal identification and law enforcement, migration, border protection, and many more. Fingerprints are easy to use because they are simple to obtain and have almost no cost to verify.

Germany's Chaos Computer Club once wanted to show how insecure biometrics are, especially a biometric that can be transferred or taken with simple physical touches. They printed 4,000 copies of the most recent edition of their own magazine. Then it was printed two ways: one by using traditional ink on paper, and the second by using a film of flexible rubber that contains partially dried glue. The second one can capture an individual's fingerprints. Later, they successfully captured the fingerprint of Wolfgang Schauble, Germany's interior minister, from his right index finger because Schauble is a right-handed person [32]. In 2013, The Guardian reported on the iPhone 5S fingerprint sensor, which was also hacked by Chaos Computer Club. That hacker group created a fake fingerprint using thin film and used it to unlock the iPhone 5S. For this hacking attempt, they printed a high-resolution fingerprint image of a user, printed it by laser on thin film, covered it with wood glue, and attached it to a real finger [33]. BBC published a report on car thieves who steal fingerprints in Malaysia [34]; Sky News reported that hospital doctors used fake fingerprints to check-in their absent colleagues [35]; CNN published a report in 2010 that a passenger boarded an Air Canada flight disguised as an elderly man [36] [37]. A group of white-hat hackers reported that it is easy to hack the fingerprint lock on the Samsung Galaxy 5 [38]. In March 2016, A group of researchers from Michigan State University successfully fooled the fingerprint sensors of a Samsung Galaxy S6 and a Huawei Honor 7 using printed fingerprints [39]. Bontrager generated fingerprints using a Generative Adversarial Network (GAN) to unlock the fingerprint recognition system [40].

2.5 Fingerprint Presentation Attacks Detection

Fingerprint Presentation Attacks Detection (PAD) is classified into two ways such as hardware-based and software-based.

Hardware-Based PAD Methods: Hardware-based methods include temperature, pulse oximetry, skin resistance, and electrical conductivity, which check the features that distinguish live humans. Coli et al. suggested an optical capture device-based method [41]. Darlow et al. [42] proposed an internal fingerprint zone detection-based model using 3D Optical Coherence Tomography (OCT) fingertip scans. Goicoechea-Telleria et al. presented two low-cost handheld microscope models with special lighting conditions [43]. Hammad et al. proposed a multimodal biometric system by fusing electrocardiogram (ECG) [44]. Keilbach proposed a laser speckle contrast imaging (LSCI) based method that is focused on liveness features such as blood flow [45].

Software-Based PAD Methods: Software-based methods utilize different image processing algorithms and programs (such as handcrafted features based Machine Learning approaches, Deep Learning-based approaches). These methods are being used to extract dynamic and static features. Dynamic features include Ridge distortion and perspiration distortion. Static features include things like perspiration distortion, texture features, and pore based methods. Some of the state-of-the-art methods are Abhyankar et al. 2006 [46], Galbally et al. [47], Gottschlich et al. 2014 [48], Goicoechea-Telleria et al. 2019 [49].

Fingerprint PAD has been shown in many procedures to be used to prevent fingerprint presentation attacks. Fig. 2.4 highlights a summary of the methods that researchers came up

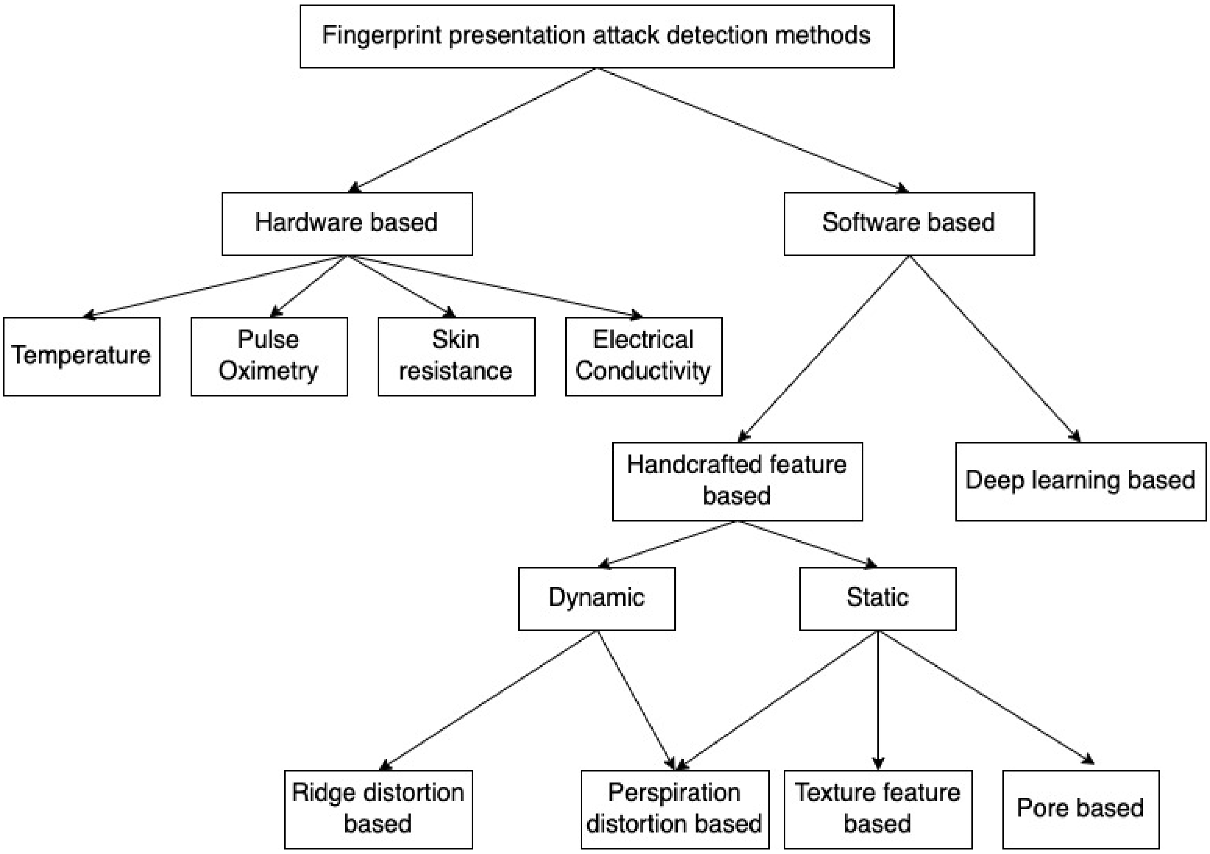


Figure 2.4: Types of existing fingerprint spoofing methods with for detecting presentation attacks [50] [51].

2.6 Feature Extraction

Feature extraction holds significant importance in image processing and computer vision applications. It is a key component of the dimensionality reduction technique, in which an initial set of raw data is divided and compressed into more manageable groups. In simple words, for an image, each pixel indicates a unit of data, and image processing extracts only useful information from the image, which minimizes the entire data amount while preserving the pixels that indicate the important features of the image. The process entails converting unprocessed image data into a condensed and significant depiction, capturing fundamental patterns, textures, or shapes that are inherent in the images. These extracted features can then be applied to different tasks such as object recognition (i.e. face recognition), image classification (i.e. real or fake images), and image retrieval. Color, shape, and texture are some of the main image features.

2.6.1 Color

Many color spaces, sometimes referred to as color models, exist to represent digital images, and each color space has a unique set of applications. Color spaces are essentially color classification systems. There are multiple color spaces for an image, including RGB, HSV, Grayscale, YCrCb, CMYK, etc. The following is an explanation of each color space:

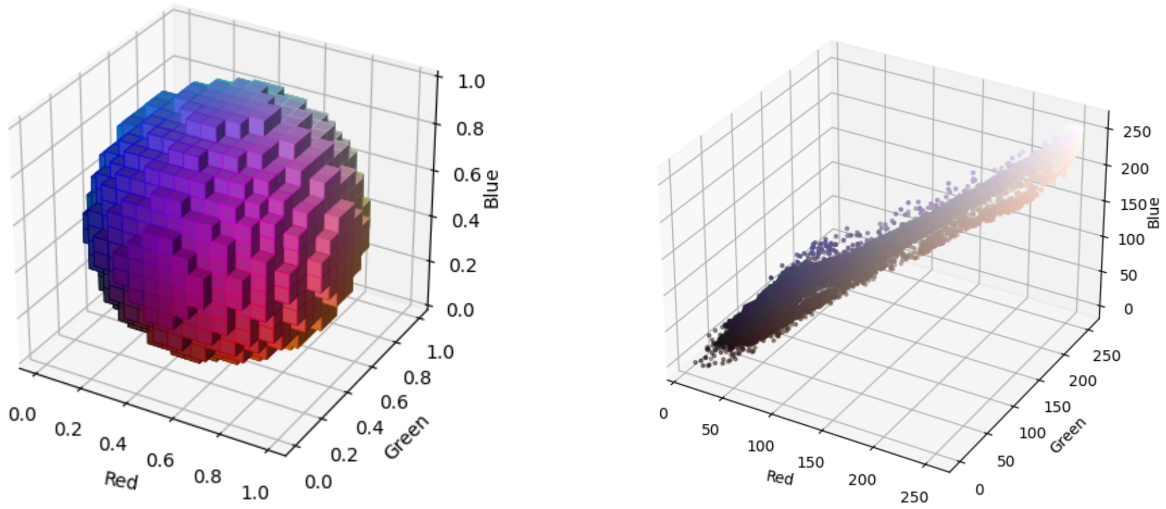
RGB (Red, Green, Blue): Within all color spaces, RGB [52] is the most widely utilized color space, which is represented by the initial letters of its components. The RGB color space has three channels, or components — red, green, and blue — each with 256 (2^8) steps and generates 16,777,216 (256 * 256 * 256) distinct color combinations. Most computer displays, digital cameras, scanners, projectors, tablets, smartphones and televisions use the RGB color space. However, RGB is a device-dependent color space. Fig. 2.5a presents a sample of RGB color space and Fig. 2.5b presents the RGB color space for a sample facial image.

HSV (Hue, Saturation, Value): HSV [53] color space divides color information into three components: Hue, Saturation, and Value. Hue indicates the type of color, Saturation indicates the intensity or purity of the color, and Value indicates the brightness or lightness of the color.

Grayscale: Grayscale [52] is a single-channel color space in which each pixel value indicates the intensity or brightness of the original image's corresponding color pixel. Grayscale images use different shades of gray to present visual content.

YCrCb: Color information for the YCrCb color space [54] is divided into luminance (Y) and chrominance (Cr and Cb) components. Y indicates brightness, while Cr and Cb reflect color differences.

CMYK: Color printing and design generally use the CMYK color space [54]. It is a combination of four colors: Cyan (C), Magenta (M), Yellow (Y), and Key (K) (expressing black).



(a) Sample RGB colors

(b) A sample facial image

Figure 2.5: 3D volumetric plot for sample RGB colors and a sample facial image

2.6.2 Shape

Various image processing techniques can be used to extract the various shapes contained within an image. These shapes are important for object recognition, image segmentation, and pattern recognition. Lines, Edges [55], Contours, Circles, Ellipses [56], Rectangles, and Polygons are some of the most frequently extracted shapes.

2.6.3 Texture Feature

Image texture features are a type of image descriptor that describes the distribution of intensity values or color patterns in an image. Texture features play important roles in several Image Processing and Computer Vision tasks, such as texture classification, s egmentation, a nd recognition. Popular texture feature extraction approaches includes Local Binary Patterns (LBP) [57], Histograms of Oriented Gradient (HOG) [58], Gabor Filters [59], Gray-Level Co-occurrence Matrix (GLCM) [60], Gray-Level Run-Length Matrix (GLRLM) [61], and Local Phase Quantization (LPQ) [62].

2.6.3.1 Local Binary Patterns. In 1994, Ojala et al. [57] proposed LBP for 2D texture patterns; it is a gray-scale invariant formed by thresholding the values of a 3 x 3 neighborhood with respect to its center pixel; if the center pixel value is greater than or equal to its neighboring pixel at that time, it is placed as 1, and if it is less than, it is placed as 0. Fig. 2.6 illustrates an LBP calculation for a facial image. Then, the LBP pattern for a selected pixel (x,y) derived from $S^{(i)}$ can be expressed as [63]:

$$LBP_{P,R}^{(i)}(x,y) = \begin{cases} \sum_{p=0}^{P-1} s(g_p^{(i)} - g_c^{(i)}) * 2^n & if\, U^{(i)} \leq 2 \\ P(P-1)+2 & otherwise \end{cases} \tag{2.1}$$

where,

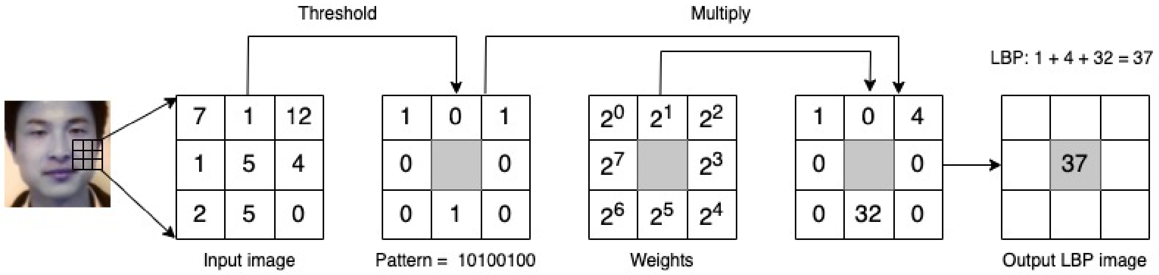
$$U^{(i)} = |s(g_{P-1}^{(i)} - g_c^{(i)}) - s(g_0^{(i)} - g_c^{(i)})| + \sum_{p=1}^{P-1} |s(g_p^{(i)} - g_c^{(i)}) - s(g_{p-1}^{(i)} - g_c^{(i)})| \tag{2.2}$$

g_c and $g_p (p = 0, 1, ..., P-1)$ refer respectively to the value of the center pixel (x,y) and the values of P equally spaced pixels on a circle of radius $R(R > 0)$, $U^{(i)}$ is for uniform LBP, and s is a thresholding function, which is defined as follows:

$$s(x) = \begin{cases} 1, & if\, x \geq 0; \\ 0, & if\, x < 0 : otherwise \end{cases} \tag{2.3}$$

It is possible to calculate LBP in a single scan through the input image. The LPB operator has been improved by taking into consideration different neighborhood sizes [63]. For example, the operator $\mathrm{LBP}_{4,1}$ uses only 4 neighbors on a circle of radius 1, while $\mathrm{LBP}_{8,2}$ considers the 8 neighbors on a circle of radius 2. In general, the operator $\mathrm{LBP}_{\mathrm{P,R}}$ refers to a number of neighboring pixels (P) on a circle of radius (R) that form a circularly symmetric neighbor set. Fig. 2.7 indicates different types of neighborhood sets.

According to the 2^P different binary patterns that the P pixels in the neighbor set can cre-



7	1	12
1	5	4
2	5	0

1	0	1
0		0
0	1	0

2^0	2^1	2^2
2^7		2^3
2^6	2^5	2^4

1	0	4
0		0
0	32	0

	37	

Figure 2.6: Illustration of an LBP calculation for a facial image.

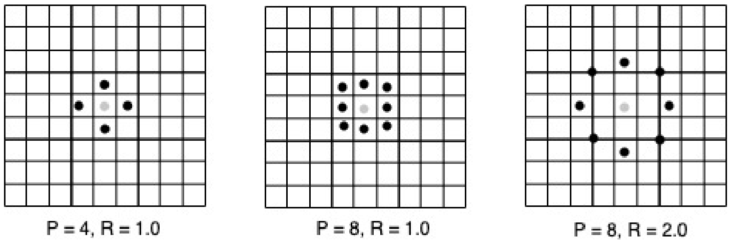
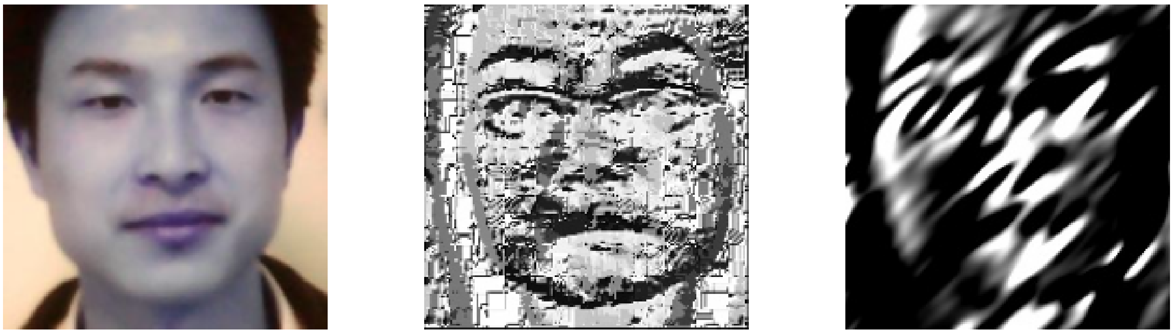


Figure 2.7: Neighborhood set for different *P* (number of neighboring pixels) and *R* (radius).

ate, $\mathrm{LBP}_{\mathrm{P,R}}$ generates 2^P different output values. There is evidence that certain bins contain more information than others. A larger number of pixels results in a larger number of labels being produced, which increases the size of the histogram feature vector in addition to the computational complexity of its calculation. Fewer labels may cause a loss of important information [63]. Therefore, to describe textured images, it is possible to use only a subset of 2^P Local Binary Patterns.

Two types of LBP patterns exist: uniform and non-uniform. A binary code is uniform if it has no more than two transitions from 0 to 1 or 1 to 0. 11000001 (2 transitions), 00110000 (2 transitions), and 11110000 (1 transition) are examples of uniform patterns, but 01010110 (6 transitions) or 10101100 (5 transitions) are non-uniform. Fig. 2.8b presents a LBP facial image and Fig. 2.9b presents a LBP fingerprint image.

2.6.3.2 Gabor Filter. The Gabor function has been established as an important tool in the Computer Vision and Image Processing domains, particularly for image texture analysis, because of its optimal lo-calization properties in both the spatial and frequency domains. After Gabor et al. [64] first proposed the 1-D Gabor function, a large number of publications relating to its applications



(a) Normal image (b) LBP image (c) Gabor filtered image

Figure 2.8: Facial image at normal version, LBP image and Gabor Filtered image

have been published. Later, Daugman et al. [65] introduced the 2-D Gabor Filter to understand the orientation-selective and spatial-frequency-selective receptive field properties of neurons in the brain's visual cortex, which he [59] then further mathematically explained. The following is an overview of a complex 2-D Gabor Filter over the image domain (x,y) [65], [59], [66].

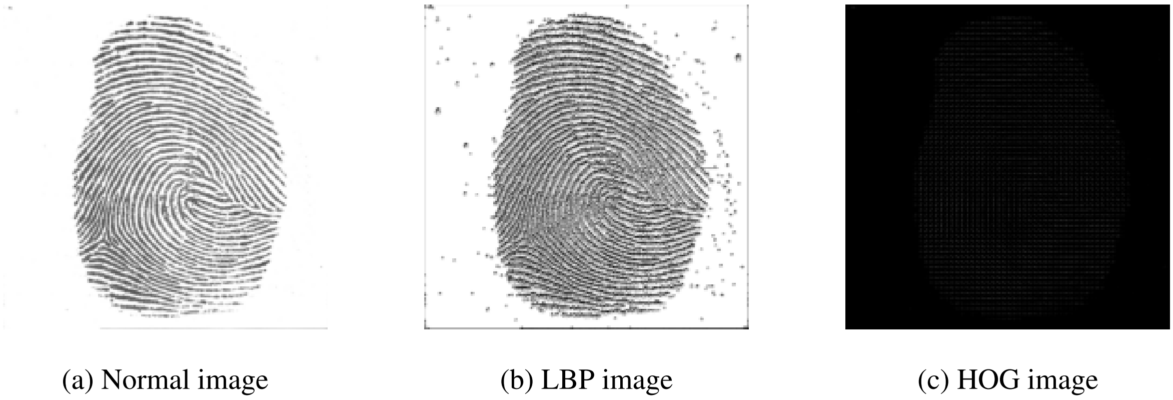
The convolution kernel is defined as:

$$G(x,y,\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\exp\left[i\left\{2\pi\frac{x'}{\lambda}+\psi\right\}\right] \tag{2.4}$$

where $x' = xcos\theta + ysin\theta$ and $y' = -xsin\theta + ycos\theta$. In Eq. 2.4 λ = the wavelength of the sinusoidal component, θ = Conveys the positioning of the parallel stripes on the filter, ψ = the offset angle of the sinusoidal function, σ = the standard deviation of the Gaussian envelope, γ = the spatial aspect ratio that defines the ellipticity of the support for Gabor function.

In this study, based on these properties, we apply a Gabor Filter to detect facial spoofing. Fig. 2.8c presents a Gabor Filtered facial image.

2.6.3.3 Histograms of Oriented Gradient. Dalal et al. came up with the idea for the Histograms of Oriented Gradient (HOG) to detect humans on images with a wide range of pose variations and backgrounds [67]. HOG counts occurrences of gradient orientation in localized portions of an image. For this study, we divided a given region into 8 × 8 cells, pixel per cell. Assuming that one block consists of 2 × 2 cells,



(a) Normal image

(b) LBP image

(c) HOG image

Figure 2.9: Fingerprint image at normal image, LBP image and HOG image

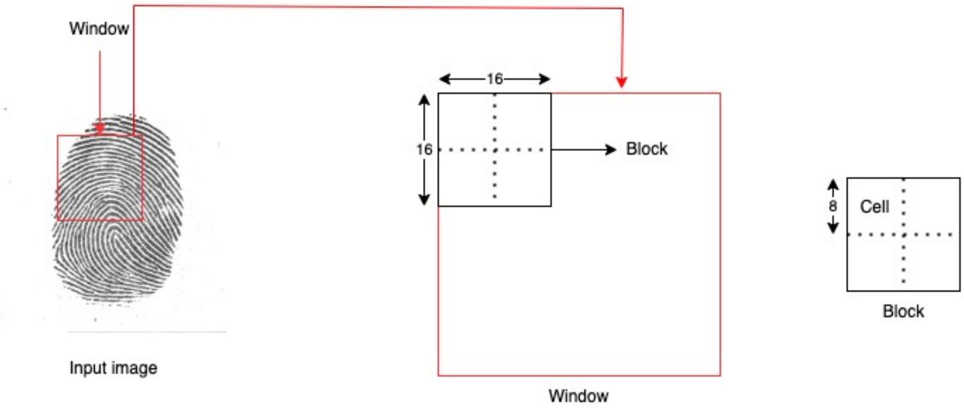


Figure 2.10: Cell and block in HOG for a fingerprint image.

which is the size of the block over which we normalize the histogram, we have a 9 x 1 matrix as orientation, which denotes the number of buckets we want to produce. Fig. 2.10 shows block, which is 2 x 2 cells, and cell, where there are 8 pixels per cell, calculation process. Fig. 2.11 presents HOG feature extraction process from a fingerprint image.

To calculate HOG, we need to calculate gradient values. Let F be a fingerprint image that is to be analyzed. Here, the norm value (G) and orientation (α) of each pixel (x,y) are calculated by the following equations [68]:

$$\begin{aligned} F(x,y) &= \sqrt{i(x,y)} \\ \text{Horizontal gradient, } G_x(x,y) &= F(x+1,y) - F(x-1,y) \\ \text{Vertical gradient, } G_y(x,y) &= F(x,y+1) - F(x,y-1) \\ \text{Norm value, } G(x,y) &= \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \\ \text{Orientation, } \alpha(x,y) &= tan^{-1}(G_y(x,y)/G-x(x,y)) \end{aligned} \tag{2.5}$$

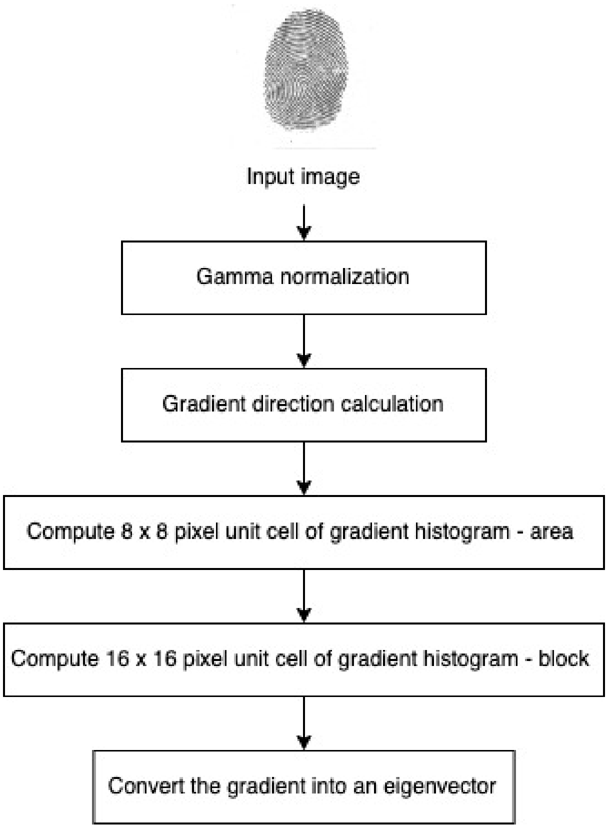

Input image
Gamma normalization
Gradient direction calculation
Compute 8 x 8 pixel unit cell of gradient histogram - area
Compute 16 x 16 pixel unit cell of gradient histogram - block
Convert the gradient into an eigenvector

Figure 2.11: Diagram for HOG feature extraction method.

The next step is to divide the image into cells and compute the orientation-based histogram in each cell. Within 0° and 180°, the orientation bins are evenly distributed. A histogram of cells is created by adding up the magnitude of the gradient for each orientation.

Dalal et al. [67] uses L2-Hys normalization in his proposed work on HOG, but during discussion, L2-Hys, L2-norm, and L1-sqrt work equally well. For our experiment, we use L2 normalization. Calculating L2-norm is as follows:

$$\text{Vector, } \upsilon(n) = \frac{\upsilon(n)}{\sqrt{1 + \sum_{k=1}^{2x2x9} \upsilon(k)^2}} \tag{2.6}$$

After moving the block by one cell, a normalized vector is computed. This is carried out for each of the 9 blocks.

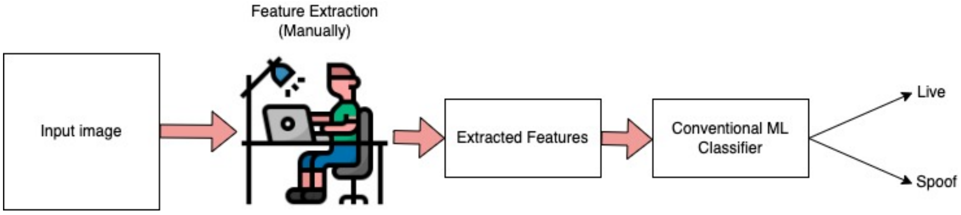


Figure 2.12: Conventional Machine Learning approach.

2.7 High Performance Computing

High Performance Computing (HPC) is the use of powerful and cutting-edge computing devices to solve complex problems that require significant amounts of memory and processing power. HPC systems are designed to do modeling at very high speeds. They are used in the fields of science, engineering, and research to do large-scale simulations, data analysis, and modeling that might not be feasible or would take a long time on conventional computers. For this work, we used UTRGV HPC because our experiment required high computing power. UTRGV HPC is funded by the National Science Foundation (grant number 2018900) as well as the Department of Defense (grant number W911NF2110169) [69].

2.8 Machine Learning

Artificial Intelligence (AI) is a field of computer science that explores how a machine, software, or system can work independently without human intervention. Machine Learning (ML) is a subset of AI. Machine Learning develops techniques and models to help computers learn from data and make predictions or decisions based on that input data. Data, model, and loss are the main three components of Machine Learning methods [70]. Supervised Learning and Unsupervised Learning are the main types of Machine Learning. Fig. 2.12 presents how the conventional Machine Learning workflow starts with manually extracting relevant features from images.

Supervised Learning: In this work, we use the Supervised Learning [71] technique. There are many state-of-the-art Supervised Learning algorithms available, including Naive Bayes, Nearest Neighbor, and Discriminant Analysis for classification problems; to address regression

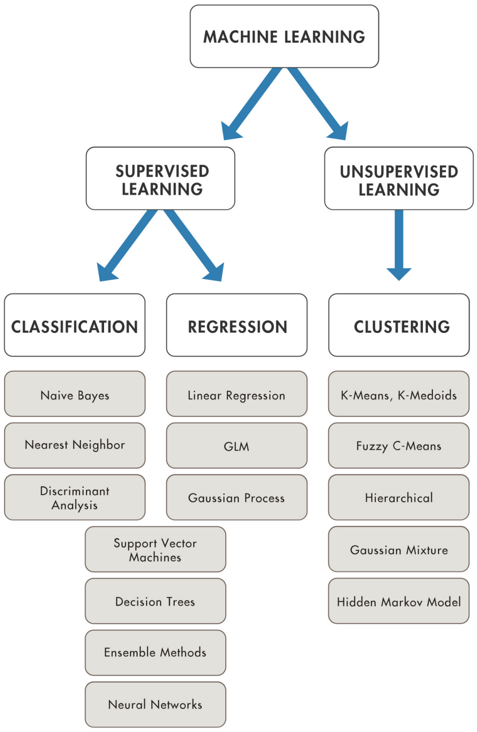


Figure 2.13: Different types of state-of-the-art Machine Learning techniques.

questions, there are linear regression, Generalized Linear Model (GLM) [72], and Gaussian Process.

Unsupervised Learning: In unsupervised learning, the machine learning model is trained using data that has not been labeled or classified in any way. Contrary to supervised learning, which utilizes labeled data to train a model, unsupervised learning employs "unlabeled" data and focuses on discovering patterns, structures, or connections within the data without any explicit guidance. Hierarchical, Gaussian Mixture, Hidden Markon Model, K-Means, K-Medoids, and Fuzzy C-Means are all ways to tackle clustering problems [73–76].

Besides these, Semi-Supervised Learning [77], [78], which is a combination of Supervised and Unsupervised Learning, and Reinforcement Learning [79] are available. Fig. 2.13 presents state-of-the-art Machine Learning techniques [73].

2.9 Deep Learning

In the 1980s, Deep Learning (DL) was first introduced. DL is a subset of Machine Learning that is based on training artificial Neural Networks (NNs) to learn and make actionable decisions

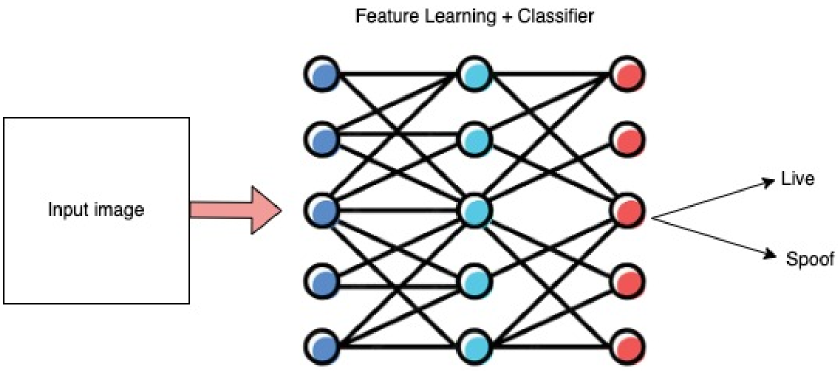


Figure 2.14: Deep Learning approach.

by handling large sets of labeled data. DL models are based on the inspiration of human brain mechanisms, where information is transported by neurons that are linked to other neurons [80]. DL needs a large amount of processing power. GPUs with huge amounts of processing power have a parallel architecture that performs effectively for deep learning.

In a DL-based approach, relevant feature extraction works automatically from images and is performed as an end-to-end learning technique where, as input data, raw data are given as inputs to a network and an output task, such as a classification problem as our PAD model, and the DL model learns how to automatically classify input data. Fig. 2.14 presents the workflow of a DL model.

2.10 Convolutional Neural Networks

Deep Learning models have many forms, such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), Transformer Networks, Reinforcement Learning models, and more. Within all models, CNNs [81] [82] are the most commonly used and popular Deep learning models, which are used for image classification, object detection, and computer vision tasks.

CNNs consist of several layers, including the input, convolutional, activation functions such as Rectified Linear Unit (ReLU), pooling, and fully connected l ayers. In CNN’s training time, it needs to adjust its parameters, such as weights and biases, to minimize a specified loss function that is based on the output task and optimization algorithms like Gradient Descent, Stochastic

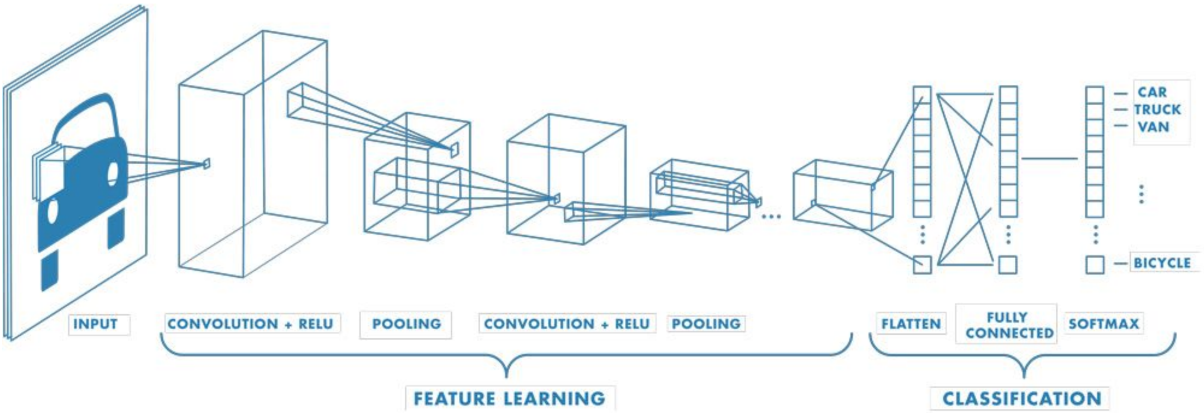


Figure 2.15: A general workflow of a CNN model network with many convolutional layers.

Gradient Descent (SGD), Adam, Adagrad, RMSProp, AdaDelta, along with other algorithms. Fig. 2.15 demonstrates a general layout of a CNN model network with many convolutional layers [83].

2.11 Transfer Learning of Pre-Trained CNN Models

Transfer learning (TL) is a well-known Deep Learning technique that involves utilizing the knowledge acquired from pre-training a model on one task and transferring that knowledge to a distinct but similar task. When dealing with CNNs, transfer learning often involves using a pre-trained CNN model on a large dataset, followed by fine-tuning on a new dataset or a specific task [84].

In our experiment, we used transfer learning techniques with pre-trained CNN models because this technique offers several advantages, such as reduced training time, better generalization for fine-tuning to the new data, improved performance, and the use of fewer computational resources [85]. Some transfer learning models (VGG, ResNet, Inception, MobileNet, and more) that have been trained to identify different features in images may be put to use in a variety of tasks.

2.12 Support Vector Machine

Support Vector Machine (SVM) [86] is a widely known supervised machine learning method to solve classification and regression-related problems. SVM finds the optimal hyperplane

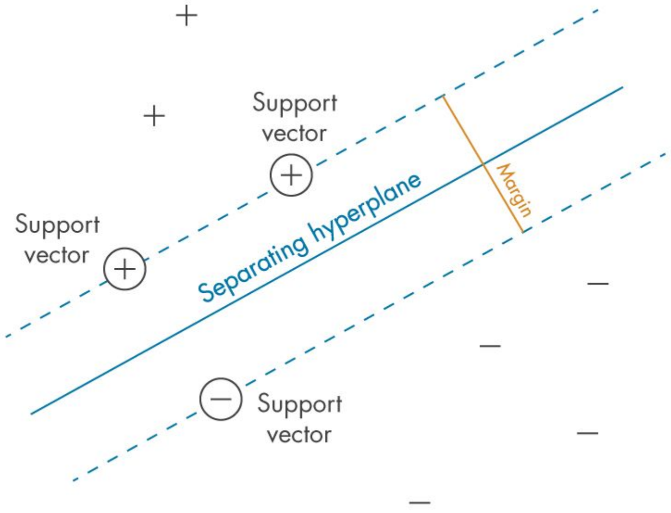


Figure 2.16: A general architecture of SVM.

that effectively separates data points from several distinct classes in the feature space. In our binary classification problem, such as figuring out whether a live or spoof image is present, this hyperplane is specifically selected to maximize the margin between the live and spoof classes. The margin refers to the gap or distance between the hyperplane and the closest data points of each class, which are called support vectors. Fig. 2.16 presents the margin and separating hyperplane of a SVM, where SVM tries to optimize its hyperplane.

Hyperplane, support vectors, margin, soft margin (C-SVM), and kernel trick are some key components and concepts of SVM [87]. However, when dealing with very large datasets, SVM can incur significant computational costs. In such scenarios, using a linear kernel SVM or alternative algorithms like logistic regression might be better. Despite this, SVM still holds its popularity for classification tasks, particularly in situations involving complex data and the requirement for high accuracy.

2.13 CNN-SVM Advantages

CNN-SVM integrates Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) for classification tasks. CNN-SVM uses a CNN to extract high-level features from the input data, and then feds these features as input for the SVM to perform the final classification task. Combining CNN and SVM leverages the strengths of both models. The CNN is responsible for learning hierarchical features from raw image data, and the SVM is responsible for detecting the optimal decision boundary to classify these features into different classes. CNN-SVM has demonstrated efficacy in a variety of image recognition tasks, particularly when the CNN is pre-trained on a large dataset and then fine-tuned for a particular task. The initial pre-training of the CNN on large-scale datasets empowers it to learn versatile features that can be beneficial in various image recognition problems. While the CNN is responsible for learning the high-level features, the SVM allows for the creation of an effective and useful classifier [88].

2.14 Advantages of the Fusion of Hand-Crafted Features and CNN

The fusion of hand-crafted features and Convolutional Neural Networks (CNNs) brings together the benefits of both conventional computer vision methods and modern deep learning approaches. This combination offers numerous advantages across diverse applications:

Fine-Tuning and Transfer Learning: Hand-crafted features can be used in transfer learning to leverage knowledge from pre-trained models and improve performance on new tasks. When you incorporate manually crafted features into the model, you improve the network's ability to handle new tasks, all the while preserving the advantages of specialized features designed by experts. (Yosinski et al. [89])

Domain-Specific Features: Hand-crafted features can capture domain-specific knowledge that may not be easily learned by deep learning models alone. Integrating these features into the model can boost its performance and align it more closely with the specific demands of the domain. (LeCun et al. [90])

Low-Resource Environments: When faced with restricted computational resources, utilizing manually engineered features alongside a scaled-down CNN architecture can offer a pragmatic approach that maintains competitive outcomes.

2.15 Federated Learning

In 2017, McMahan et al. proposed Federated Learning (FL), which is a decentralized machine learning approach. Federated learning is trained on distributed devices, such as mobile devices, and shares locally updated parameters with a central server. In Federated Learning, distributed devices are called clients. In Federated Learning, there are C_1, C_2, C_3, ..., C_k clients, and for each client, C_i has its own local dataset, D_i. The whole training process of Federated Learning can be listed in three different steps: initialization, local model training (at client devices), and global aggregation (at the server) [91].

Initialization: The server initializes the parameters of the global model. At first, the server initializes the parameters of the global model. Depending on the training problem, this can be done randomly or pre-trained on a public dataset. After this initialization, the parameter server sends the initial global model parameters ω_0 to the client devices to start their local model training.

Local Model Training: After getting the initial model, ω_0, from the server, clients start training that global model based on their own local dataset, D_i. After finishing t^{th} training rounds, clients update the local model ω_t^i based on the received initial model, ω_0. Then clients send the updated local model parameters ω_{t+1}^i to the server. For local model, depending on the FL algorithm, the loss function can be different [92]. In this experiment, for the Support Vector Machine-based FL model [92], [93], hinge loss function $l(\omega,x,y) = max(0, 1 - y.(\omega.x))$ is used as a loss function for SVM. Loss function for Federated Learning F can be written as:

$$F(\omega,X,Y) = \sum_i l(\omega,x_i,y_i) + \alpha||\omega||^2 \tag{2.7}$$

where, x_i = input or feature vector for sample, y_i = binary label, α = regularization term.

Global Aggregation: The server combines all model updates received from local clients and produces a new version of the global model as follows:

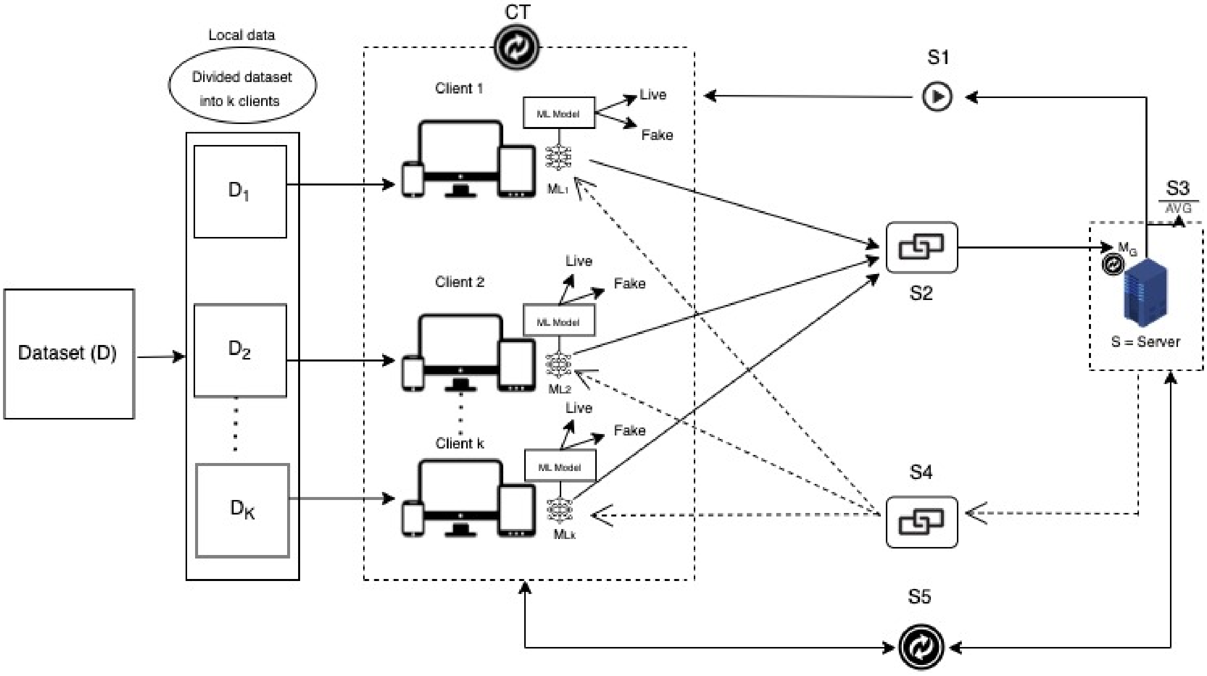


Figure 2.17: A general workflow of Federated learning

$$F(\omega_t) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{K}|\mathcal{D}_i|F(\omega_t^i), i \in 1,2,....,K \tag{2.8}$$

Iterations of the above procedures will be executed until the required level of accuracy is obtained.

FedAvg [91], FedProx [94], FedPAQ [95], FierFVG [96] are examples of Federated Learning aggregation algorithms. Fig. 2.17 presents a general workflow of Federated Learning.

Here, at fig. 2.17,

- CT = Each devices trains local model based on local data
- S1 = Server & Devices agree on model initialize parameters
- S2 = Devices (client) send local parameters to server, S
- S3 = Server aggregates & updates parameters of global model
- S4 = Server (S) sends updated global model to all devices
- S5 = Repeat until convergence

2.16 Advantages of Federated Learning

In Federated learning, there are a couple of advantages:

- **Privacy preservation:** In Federated learning, Clients don't share raw training data with the central server; they just share updated parameters with the server, so through this process, it is possible to secure sensitive privacy data and train on large amounts of data while ensuring privacy for participants. In this process, it is also possible to mitigate security risks [91].

- **Energy efficiency:** Federated learning distributes the whole training process among multiple devices, which can minimize energy consumption on individual devices compared to traditional centralized methods. This is highly beneficial for resource-constrained devices such as mobile phones and Internet of Things (IoT) devices [97], [98], [99].

- **Standard regulations:** Data protection laws, like the General Data Protection Regulation (GDPR), set limits on how personal data can be distributed and preserved in certain situations. Federated learning helps organizations maintain these regulations by keeping data on the device end and minimizing the probability that the data will be disclosed [100], [101].

- **Minimized inference latency:** Federated learning enables individual clients to continually train and update their Machine Learning models directly on their devices. The updated model can then be used to make predictions locally on the client's device. This local decision-making process results in minimal latency compared to the traditional method of making decisions on a centralized server [102].

Federated Learning is being used and liked in many different areas, such as healthcare, mobile devices, the Internet of Things (IoT), and edge computing. Fig. 2.18 presents diverse applications of Federated Learning, including the Energy sector and Social Sciences. Fig. 2.19 depicts the research activities in Federated Learning from 2017-2022, showing they are

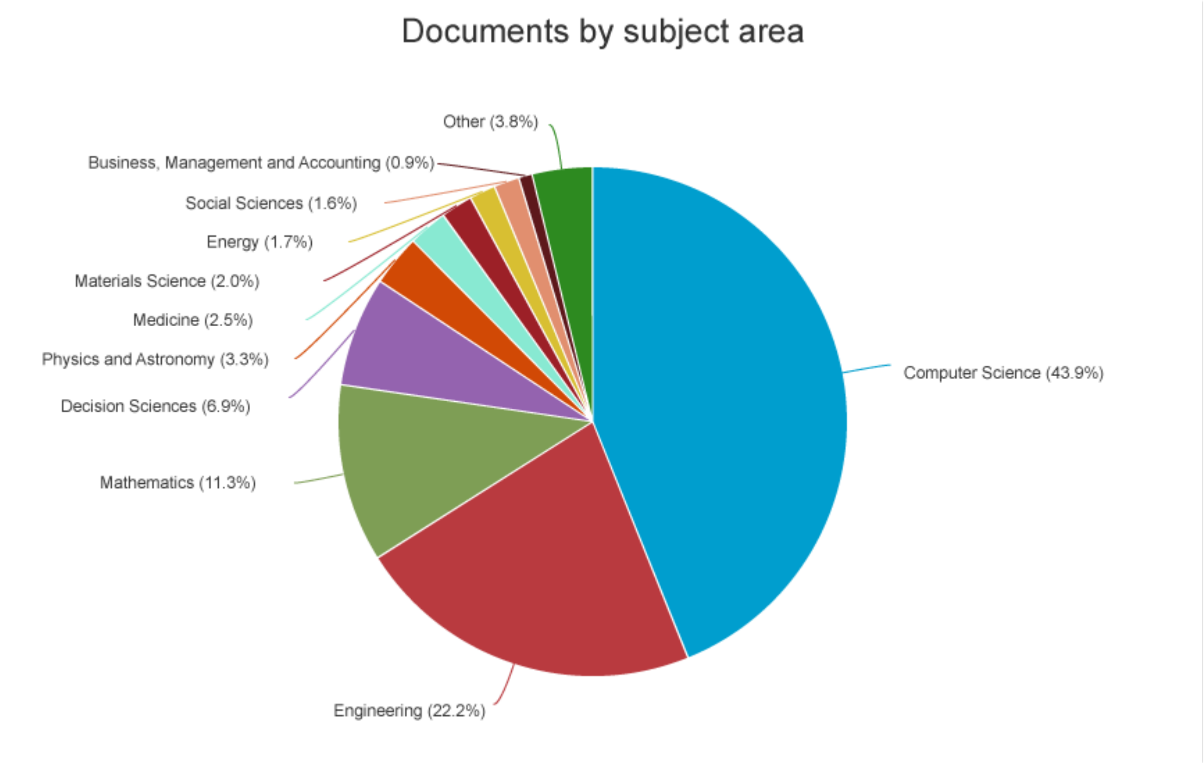


Figure 2.18: Diverse applications of Federated Learning.

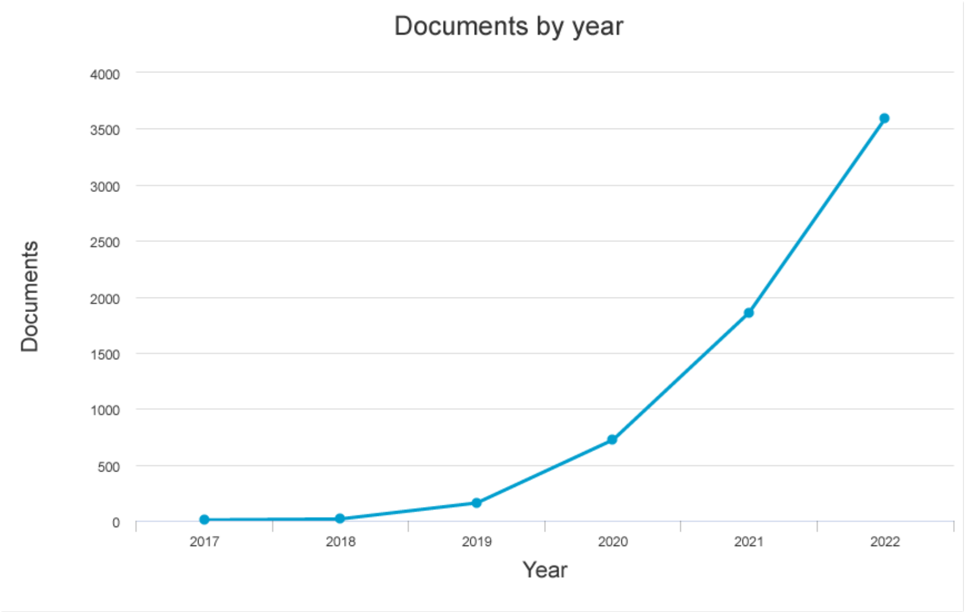


Figure 2.19: Total number of Federated Learning-related research documents in 2017–2022.

growing exponentially. While it addresses privacy issues effectively, it also formulates new problems with overhead communication costs, maintaining data consistency, and dealing with heterogeneous data among devices.

CHAPTER III

LITERATURE REVIEW

This chapter presents an in-depth review of the various state-of-the-art methods that have been established to identify presentation attacks over the years. Conventional Machine Learning, Deep Learning, and Federated Learning-based techniques are the three main categories used to categorize the currently existing methods.

3.1 Facial PAD SOTA in Federated Learning

Table 3.1 presents the most recent state-of-the-art methods for detecting facial presentation attacks in Federated learning.

Table 3.1: Table for state-of-the-art of Facial PAD in Federated Learning

Author	Anti-spoof Method	Datasets	Method	Attack Type	Evaluation Metrics[%]
Shao et al. (2022) [103]	Feature Based	FedPAD+fPAD+ FedGPAD	Oulu-NPU, CASIA-MFSD, Idiap Replay-Attack, MSU-MFSD, SiW	Print, Replay	HTER 28.19%, AUC 23.01%

Continuation of Table 3.1

Author	Anti-spoof Method	Dataset	Method	Attack Type	Evaluation Metrics[%]
Shao et al. (2021) [104]	Feature Based	Oulu-NPU, CASIA-MFSD, Idiap Replay-Attack, MSU-MFSD , SiW, 3DMAD, HK-BUMARsV2	Federated Test-Time Adaptive fPAD with Dual-Phase Privacy Preservation Framework	Print, Replay, Mask	HTER 16.97%, AUC 90.25%
Shao et al. (2020) [105]	Combination Based	Oulu-NPU, CASIA-MFSD, Idiap Replay-Attack, MSU-MFSD, SiW, 3DMAD, HK-BUMARsV2	FedPAD	Print, Replay, Mask	HTER 30.51%, EER 26.10%, AUC 84.82%
Liu et al. (2022) [106]	Deep Learning Based	CASIA-Webface, LFW, CFP-FP, AgeDB30	FedFV	Print	0.16%(LFW), 4.54%(CFP-FP) and 7.15%(AgeDB-30) [Accuracy]

Continuation of Table 3.1

Author	Anti-spoof Method	Dataset	Method	Attack Type	Evaluation Metrics[%]
Chen et al. (2022) [107]	Feature Based	NUAA, MSSPOOF, CASIA-SURF	FedFSAD	Print	0.996 ± 0.001% (NUAA), 0.998 ± 0.003% (MSSPOOF), 0.871 ± 0.005%()[ACC]
End of Table					

3.2 Facial PAD SOTA using ML and DL in Last Decade

Table 3.2 presents the most recent state-of-the-art methods for detecting facial presentation attacks over 2012–2023.

Table 3.2: Table of facial PAD in the last decade

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Pei et al. (2023) [108]	Deep Learning Based	SiW, CASIA FASD and REPLAY-ATTACK	Deep Siamese Network	Cross Presentation Attack	1.13(CASIA) [EER]

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Shu et al. (2023) [109]	Deep Learning Based	CASIA-FASD, REPLAY-ATTACK, and OULU-NPU	MSCI-DSCNN method	Print, Replay	2.9(CASIA), 4.7(REPLAY), 9.6 ± 6(OULU-NP) [APCER]
Wang et al. (2023) [110]	Deep Learning Based	REPLAY-ATTACK, CASIA-FASD, OULU-NPU, and SiW	learnable gradient operator (LGO)	Print, Replay	1.11(CASIA) [EER]
Huang et al. (2023) [111]	Deep Learning Based	SiW, OULU-NPU, CASIA-FASD, MSU-MFSD and REPLAY-ATTACK	Combination of AFD GSAL and PBMS	Print, Replay	17.78 (O&M&I to C) [HTER]
Yılmaz et al. (2023) [112]	Feature Based	NUAA, CASIA, REPLAY-ATTACK and OULU-NPU	LBP, PCA, SVM	Print, Replay	0.17(NUAA), 0.22(CASIA), 9.28(REPLAY-ATTACK)[EER]
Chang et al. (2022) [113]	Feature Based	CASIA, REPLAY-ATTACK, UVAD, OULU-NPU, SiW and Own Dataset	Face PAD Based MIQF + SVMs	Print, Replay	36.8[EER], 9.20[FRR], 23[HTER]
Fang et al. (2022) [114]	Combination Based	CASIA-MFS, MSU-MFS and OULUNPU	LBP and Hybrid CPQD	Print and Replay	48.25% [Median BPCER]

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Wang et al. (2022) [115]	Combination Based	REPLAY-ATTACK, CASIA-FASD, OULUNPU and SiW	LGO	Print, Replay	31.9%(CASIA) [EER]
Fatemifar et al.(2021) [116]	Deep Learning Based	REPLAY-ATTACK, REPLAY-MOBILE, Rose-Youtu	CNN, SVM+Motion, SVM+Gabor, Deep pixel-wise, Wavelet, Deep Learning	Print, Replay and Mask	0 (REPLAY-ATTACK) and 8.13 (RoseYoutu) [HTER]
Ebihara et al.(2021) [117]	Feature Based	NUAA, REPLAY-ATTACK, SiW, and OULU-NPU	SpecDiff+SVM	Photo, Replay	0.93 (SiW) [APCER]
Daniel et al. (2021) [118]	Feature Based	REPLAY-ATTACK	EFD + QF	Mobile, Print, High-definition	0.31[EER]
Jia et al. (2021) [119]	Feature Based	SWFFD, WFFD, 3DMAD and HKBU-MARsV1	RAN	Mask	23.34 ± 10.35(BPCER)
Zhang et al. (2020) [120]	Feature Based	REPLAY-ATTACK and CASIA-FASD	DWT-LBP-DCT with SVM	Print, Replay	0 (REPLAY) [HTER] and 5.56 (CASIA) [EER]

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
George et al. (2020) [121]	Combination Based	WMCA, MLFP and SiW-M	MCCNN (BCE+OCCL)-GMM	Print, Replay, Mask, Makeup, Partial	12.82(REPLAY) [EER]
Sun et al. (2020) [122]	Deep Learning Based	CASIA-FASD, REPLAY-ATTACK, OULU-NPU, and SiW	SAPLC, SVM	Warped Photo, cut photo, and video	2.94, 0.38, and 7.73 [ACER] in Three Protocols
Shu et al. (2020) [123]	Feature Based	CASIA FASD, REPLAY-ATTACK, REPLAY-MOBILE, and OULU-NPU	ED-LBP, SVM	Print, Mobile and High definition	0.00[EER]
Song et al. (2019) [124]	Combination Based	NUAA, REPLAY-ATTACK, CASIA and Own Dataset	SPM, SSD, SPMT+SSD	Print, Replay	SPMT+SSD 0.72%, o.05%, 0.025%[HTER]
Yu et al. (2019) [125]	Deep Learning Based	REPLAY-ATTACK, CASIA	DK+Deep+ MKL	Print, Replay	2.78%[HTER]
George et al. (2019) [126]	Deep Learning Based	WMCA	MC-CNN	Print, Replay, Mask	0.3%[ACER]

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
George et al. (2019) [127]	Deep Learning Based	Replay Mobile, OULU	DeepPixBiS, IQM-SVM, LBP-SVM	Print, Replay	0%[HTER], 0.42%[ACER]
Chen et al. (2019) [128]	Deep Learning Based	CASIA-FASD, REPLAY-ATTACK and OULU-NPU	FARCNN+HS-Retinex-YCbCr	Print, Replay	0.062[EER]
Chen et al. (2019) [129]	Deep Learning Based	CASIA-FASD, REPLAY-ATTACK and OULU	TSCNN, MobileNet (1024D), ResNet-18 (512D)	Print, Replay	0.177(REPLAY)
Li et al. (2018) [130]	Feature Based	CASIA-FASD, REPLAY-ATTACK	Colour LBP, SVM	Print, Replay	6.2(CASIA)
Li et al. (2018) [131]	Combination Based	Idiap REPLAY-ATTACK, CASIA FAS and MSU MFSD, Rose-Youtu	CoALBP, LPQ ,SVM	Print, Replay and Mask	27.7 [HTER]

Continuation of Table 3.2					
Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Li et al. (2018) [132]	Deep Learning Based	REPLAY-ATTACK, CASIA Face AntiSpoofing, MSU mobile, RoseYoutu Face Liveness Detection Database	3D CNN	Printed Paper, Video dDisplay, Mask and Video Replay	28.7[HTER]
Li et al. (2018) [133]	Combination Based	Relay-Attack and CASIA-FA	MLBP, SVM	Replay, Print, Displayed Image	2.8(REPLAY-ATTACK)
Xiong et al. (2018) [134]	Feature Based	CASIA, REPLAY-ATTACK, MSU and Oulu Dataset	UPAD(GMM, RBF OC-SVM and AE), NN	Print, Replay	0.00[Video] [APCER]
Manjani et al.(2017) [135]	Deep Learning Based	REPLAY-ATTACK, CASIA-FASD, 3DMAD, UVAD and SMAD	DDGL+SVM	Mask	0.0, 1.3, 0.0, 16.5, 13.1 [HTER]
Chan et al. (2017) [136]	Software and Hardware Based	FaceLiveFlash	SVM	Print, Replay, Mask (2D, curved)	0.0 [HTER]

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Arashloo et al.(2017) [137]	Feature based	CASIA, REPLAY-ATTACK and MSU	LBP-TOP, LPQ-TOP and BSIF-TOP	Print, Mobile, Replay	70.23 [AUC]
Souza et al. (2017) [138]	Combination Based	NUAA	LBPnet, n-LBPnet	Print	0.021,0.018 [EER]
Peng et al (2017) [139]	Feature Based	MSU MFSD, CASIA FASD, REPLAY-ATTACK and REPLAY-MOBILE	GS-LBP, LGBP	Photo, Video	5.10 (MSU MFSD)for LGBP [EER]
Boulkenafet et al. (2016) [140]	Feature Based	CASIA FASD, MSU MFSD, REPLAY-ATTACK	HSV+YCbCr,	Print and Video	3.2(CASIA), 3.5(MSU MFSD), 0.0(REPLAY) [EER]
Phan et al. (2016) [141]	Feature Based	REPLAY-ATTACK, CASIA-FASD, and MSU MFSD	LDP-TOP, LBP-TOP	Print, Mobile, High-def	2.50(REPLAY), 8.94(CASIA), 6.54(MSU MFSD) [EER]
Li et al. (2016) [142]	Image Quality Based	REPLAY-ATTACK, CASIA FASD	IQA, M-SVR	Print	13.3 [EER

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Ali et al. (2016) [143]	Feature Based	Own Dataset	GS (Colocation, Collinearity), k-NN, SVM, LDC	Photograph, Mask and video Replay	0.05 [FPR]
Siddiqui et al.(2016) [144]	Texture and Motion Based	CASIA-FASD, 3DMAD and MSU-MFSD	SVM	Print, Replay, Wrap and Mask	3.14(CASIA), 0(3DMAD), and 0(MSU-MFSD) [EER]
Pinto et al. (2015) [145]	Feature Based	REPLAY-ATTACK, CASIA, UVAD, 3DMAD	BoVW,PLS, SVM	Print, Replay and Mask	29.87[HTER]
Menotti et al.(2015) [146]	Deep Learning Based	Biosec, Warsaw, MobBIOfake, REPLAY-ATTACK, 3DMAD, LivDet2013	SVM	Print, Replay	0.75(REPLAY- ATTACK), 0.00 (3DMAD) [EER]
Di Wen et al. (2015) [147]	Feature Based	REPLAY-ATTACK, CASIA FASD (H protocol) and MSU MFSD	IDA+SVM	Print, Replay	5.82[EER]

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Boulkenafet et al. (2015) [148]	Feature Based	REPLAY-ATTACK, CASIA	Linear SVM	Print, Replay	0.4[EER]
Tirunagari et al. (2015) [149]	Feature Based	PRINT-ATTACK, REPLAY-ATTACK, CASIA-FASD	DMD, LBP, SVM	Print, Replay	0.0(PRINT), 3.75(REPLAY), 21.75(CASIA) [HTER]
Anjos et al. (2014) [150]	Feature Based	PHOTO-ATTACK	OFC	Print, Replay	1.52[EER]
Raghavendra et al. (2014) [151]	Feature Based	3DMAD	BSIF+SVM	Mask	0.03% [HTER]
Galbally et al.(2014) [152]	Image Quality Based	REPLAY-ATTACK, CASIA-FASD	IQM	Print, mobile, Replay	17.9% [FFR]

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Galbally et al.(2014) [153]	Software Based	ATVS-FIr DB, CASIA-IrisV1, WVU-Synthetic Iris DB, LivDet 2009 DB, REPLAY-ATTACK DB	IQA	Iris, Finger-Print, Print, Replay, Highdef	0.0%[FFR]
Erdogmus et al.(2014) [154]	Combination Based	Morpho Database, 3DMAD	LBP, LDA, SVM	Mask	7[EER]
Bharadwaj et al. (2013) [155]	Motion Based	PRINT-ATTACK, REPLAY-ATTACK	LBP, HOOF	Print, Replay	0% and 1.25% [HTER]
Pereira et al.(2013) [156]	Feature Based	REPLAY-ATTACK, CASIA-FASD	Correlation, LBP-TOP, LBP	Print, Replay	54%[HTER]
Erdogmus et al.(2013) [157]	Feature Based	3DMAD	LBP,LDA	Mask	0.95%, 1.27% [HTER]
Lai et al. (2013) [158]	Spatial and Temporal Feature Based	Own Dataset	DFR	Print, Replay	0.05%[FAR]

Continuation of Table 3.2

Author	Anti-spoof Method	Dataset	Method	Attack Type	Performance Metrics[%]
Wang et al.(2013) [159]	Motion Based	Own Dataset	3D Sparse Structure Method, CLM +SVM	Print	100% Detection Accruracy
Jukka et al.(2012) [160]	Feature Based	NUAA	Multi-scale LBP+SVM	Print	4.4% [FRR]
Anjos et al.(2012) [161]	Motion Based	PRINT-ATTACK	MLP	Hand Craft Based	11% [HTER]
Jukka et al.(2012) [162]	Feature Based	NUAA, PRINT-ATTACK, Yale Recaptured	LBP, HOG, Gabor-Wavelets + Linear SVM	Print	1.1[EER]
Maria De et al.(2012) [163]	Motion Based	HONDA, NUAA	Geometric Invariants	Replay	0.0 [EER]
Chingovska et al. (2012) [164]	Feature Based	REPLAY-ATTACK	tLBP+dLBP+mLBP, LDP, SVM	Print, Replay	4.23%[HTER]

End of Table

3.3 Fingerprint PAD SOTA in Last Decade

Table 3.3 presents the most recent state-of-the-art methods for detecting fingerprint presentation attacks over 2008–2023.

Table 3.3: Table for the state-of-the-art of fingerprint PAD

Author	Anti-spoof Method	Dataset	Method	Spoofing Materials	Performance Metrics[%]
Khan et al. (2023) [165]	Feature Based	SH-DB-MOLF, CM-DB-MOLF,LV-DB-MOLF.DBNIST-302, DB-NIST-302, MOLF	FEOG	Photo	143.434[MSE], 978.20[PSNR], 0.81[Timetaken], 0.9180[SSIM], 354.74[IEF]
Abdullahi et al. (2022) [166]	Combination Based	LivDet 2013, LivDet 2015	FinSpoofNet	Ecoflex, Gelatin, Latex, Modasil, Wood Glue, Liquid Ecoflex, RTV	0.31%(LivDet 2013), 2.26%(LivDet 2015) [ACE]

Continuation of Table 3.3					
Author	Anti-spoof Method	Dataset	Method	Spoofing Materials	Performance Metrics[%]
Saguy et al.(2021) [167]	Software Based	Own Dataset	NFIQ	PEG, Silicone, Polyurethane, Latex	21%[FRR]
Pałka et al.(2020) [168]	Deep Learning and Frequency Feature Based	Own Dataset	TDS	Silicone, Latex, Plasticine, Gelatin, Play-Doh	87.9%[TDR], 3.9%[FDR] (Tim Frequency)
Arora et al.(2020) [169]	Deep Learning Based	FVC2006, ATVSFFpDB, Spoofing-Attack Finger Vein, LivDet 2013, LivDet 2015	CNN	synthetic generator SFinge, Photo, Latex, Ecoflex, Wood glue, Body Doubles, liquid Ecoflex, RTV	99% (all the benchmarks) [accuracy
Souza et al.(2019) [170]	Deep Learning Based	Crossmatch	Deep Boltzmann Machines	Photo	20.70(FAR), 8.96(FRR), 85.82(ACC)

Continuation of Table 3.3

Author	Anti-spoof Method	Dataset	Method	Spoofing Materials	Performance Metrics[%]
Toosi et al.(2019) [171]	Deep Learning Based	LivDet 2011, LivDet 2013	DBN+Spofnet, Transfer Learning (AlexNet-BN and VGG-19)	Mold	23.3% [Accuracy] (VGG-19 on LivDet2011)
Zhang et al.(2019) [172]	Deep Learning Based	LivDet2013, LivDet2015	Slim-ResCNN	Ecoflex, Gelatin, Latex, Modasil, Wood Glue, Liquid Ecoflex, RTV	95.25% [Accuracy]
Souza et al.(2017) [173]	Deep Learning Based	Crossmatch	Deep Boltzmann Machine (DBM)	Bodydouble, Playdoh, Wood Glue, Latex	19.40[FAAR], 9.76[FRR], 85.96[ACC]
Yuan et al.(2017) [174]	Deep Learning Based	LivDet (2013), LivDet (2011)	CNN, PCA	Photo, Ecoflex, Gelatin, Latex, Modasil, Wood Glue	4.57% [ACE

Continuation of Table 3.3

Author	Anti-spoof Method	Dataset	Method	Spoofing Materials	Performance Metrics[%]
Balaji et al.(2016) [175]	Feature Based	Own dataset	SIFT	Photo	40% (Finger from different person) [Matching accuracy]
Wild et al.(2016) [176]	Learning Based	Face: Idiap Replay-Attack, CASIA Face Anti-Spoofing, Fingerprint: Fingerprint Liveness Detection Competition 2013	1 Median Filtering	Photo, Video, Ecoflex, Gelatin, Latex	22% [EER]
Park et al.(2016) [177]	Deep Learning Based	LivDet2009	CNN	Gelatin, Play-doh, Silicone	3.42% [ACE]
Menotti et al.(2015) [178]	Deep Learning Based	Iris: Biosec, Warsaw, Mob-BIOfake, Face: Replay-Attack, 3DMAD, Fingerprint: LivDet2013	Spoofnet	Photo, Video	99.84%(Iris) [ACC], 92.09%(Fingerprint)[ACC]
Rattani et al.(2015) [179]	Feature Based	LivDet 2011	Weibull-calibrated SVM (W-SVM)	EcoFlex, Latex, Gelatine, Silgum, WoodGlue	97.3%[EER]

Continuation of Table 3.3

Author	Anti-spoof Method	Dataset	Method	Spoofing Materials	Performance Metrics[%]
Akhtar et al.(2014) [180]	Software Based	**Face:** Print Attack, Replay Attack, NUAA Photograph Imposter Database, Personal Photo Attack, Yale Recaptured Database, **Iris:** ATVS-FIr DB, ATVS-FIr DB, **Fingerprint:** ATVS-FFp DB, LivDet09, LivDet11-Sagem, LivDet13-Swipe	LUCID	Face: Photo, Video, Iris: Photo, Contact Lenses, Fingerprint: Silicone, Gelatine, Playdoh	1.54%(NUAA), 0.07% (Notre Dame) [HTER]
Akhtar et al.(2014) [181]	Software Based	**Face:** Print Attack, NUAA Photograph Imposter Database, **Iris:** ATVS-FIr DB, **Fingerprint:** ATVS-FFp	MoBio LivDet	Face: Printed Photo, Video, Iris: Photo, Fingerprint: Silicone	1.03%(Iris), 2.88% (Print Attack),1.54% (NUAA) [HTER]

Continuation of Table 3.3

Author	Anti-spoof Method	Dataset	Method	Spoofing Materials	Performance Metrics[%]
Coli et al.(2008) [41]	Featured Based	Own Dataset	Static and Dynamic features using Optical Sensor	Liquid Silicon Rubber	65.34(SF3 subset) [Accuracy]

End of Table

CHAPTER IV

FACIAL PRESENTATION ATTACK DETECTION

4.1 Dataset

Facial spoofing datasets are developed from videos or images that are used to investigate facial spoofing problems. This involves trying to trick a facial recognition system by displaying a fake version of a real person's face. These datasets commonly contain both real facial images and different types of fake images like photos, videos, or 3D masks.

For this experiment, we used the NUAA Photograph Imposter Database [182], which was developed by Nanjing University of Aeronautics and Astronautics, China. Fig 4.1 presents a snapshot of the NUAA Dataset. They developed this database for public access and adopted a generic webcam to capture photos. This dataset covered 15 subjects in two sessions, mainly participants who participated in that study. For the fake image, they printed photos on 70-gram A4-size paper. In the first session, they included 889 images as a training set, and in the second session, they included 854 images. So in total, they added 1743 images from 9 subjects as a training set. For imposter or fake images, they added 855 and 893 images in the training

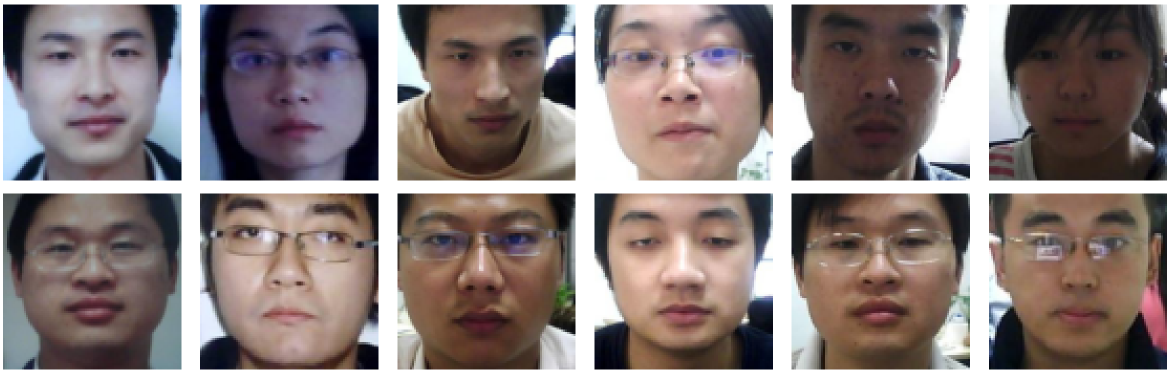


Figure 4.1: Do you want to differentiate between a live and fake face from these samples? Please try! Was it hard? So from this figure, you could assume the difficulty level of this problem. Answer: The leftmost two columns are fake faces.

Table 4.1: The total number of images available in both the training and testing datasets.

	Training Set	Test Set	Total
Live (Client)	1,743	3362	5105
Fake (Imposter)	1748	5761	7509
Total	3491	9123	12614

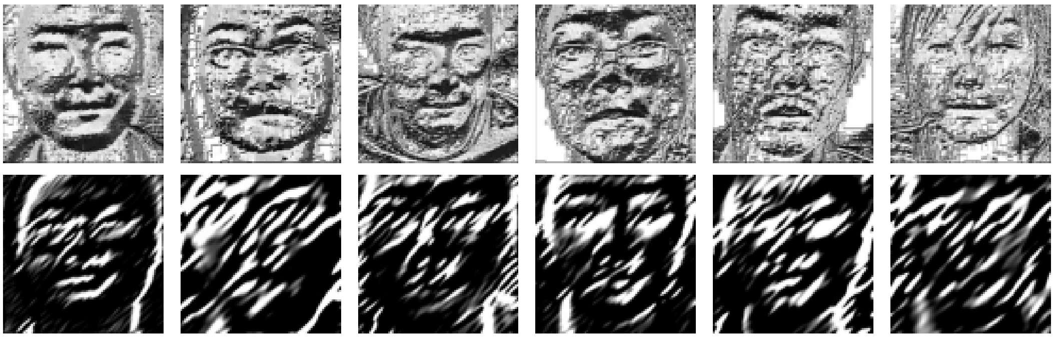


Figure 4.2: Used same images of Fig 4.1. Row 1: Constructed LBP images from input images; Row 2: Constructed images with the Gabor filter

set, respectively, from the first and second sessions. In the testing set, they added 3362 images from live humans and generated 5761 images as imposters or fakes. The overview of the NUAA dataset is presented in Table 4.1.

4.2 Feature extraction

Local Binary Pattern: For this experiment, we used Local Binary Pattern (LBP), which is the most popular texture descriptor in computer vision and image analysis. Especially in facial spoofing detection methods, most state-of-the-art techniques adopt the LBP method as a feature extrator. Ojala et al. introduced LBP for 2D texture patterns in 1994. LBP is particularly useful for different types of problems, such as face recognition, facial presentation attack detection, texture classification, and object detection.

In general, the LBP has become an important technique for investigating textures in computer vision because it is straightforward to use, works quickly, and can extract specific patterns and textures from images. Section 2.6.3.1 discusses the specifics of LBP.

Gabor Filter: Along with LBP, we also used the Gabor Filter, which is a type of linear filter that is used in image processing and computer vision for investigating and enhancing

(a) LBP facial image

(b) Gabor Filtered facial image

Figure 4.3: Processed facial image at LBP version and Gabor Filtered version.

images. A Gabor filter has several key characteristics and components, such as Frequency and Orientation Selectivity, Sinusoidal Component, and Gaussian Envelope.

Gabor filters have been applied to various tasks, including face recognition, fingerprint recognition, object detection, and texture analysis. Gabor filters can target specific patterns in an image by adjusting parameters like frequency, orientation, and scale. Section 2.6.3.2 goes over the technical aspects of the Gabor Filter in more detail.

The LBP version (Fig. 4.3a) and Gabor Filtered version (Fig. 4.3b) of a processed facial image are shown in Fig 4.3.

4.3 Model

In this experiment, we used different types of hybrid CNN-SVM models, such as ResNet-SVM, MobileNetV2-SVM, and VGG16-SVM. Here, we present the following architecture in detail:

4.3.1 ResNet-SVM

He and his colleagues [183] came up with the idea of residual neural networks in 2016. The COCO object detection dataset gained a 28% relative improvement using their methods. ResNet50 is a 50-layer deep neural network, and some of the main features of ResNet50 are skip connections or shortcut connections like adding the original input to the output of the convolutional block and batch normalization after every convolution layer.

ResNet-50 uses a bottleneck architecture in which each layer in a residual block is structured into convolutions that are 1x1, 3x3, and 1x1. Having fewer parameters and less computational load makes the network more efficient, and this design helps achieve that. After

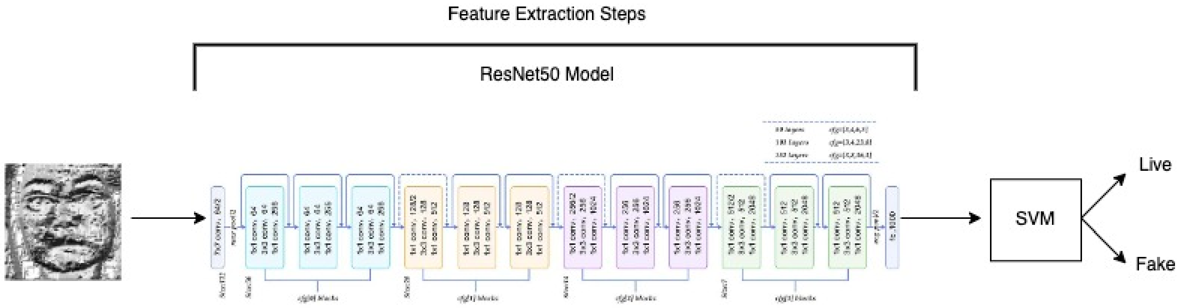


Figure 4.4: General architecture of ResNet-SVM model.

ResNet-50 was developed, there were a lot of modifications and enhancements introduced to neural network designs. ResNet-18, ResNet-34, ResNet-101, and ResNet-152 are all in the same series as ResNet-50. Each of these models features different depths and complexities.

In this experiment, ResNet50-SVM [184] was used. The last layer in ResNet-50 was taken out and replaced with an SVM to improve the model's ability to distinguish between real and fake faces. L2 regularization was used as a lasso regression method, and hinge loss was used as a loss function. Fig. 4.4 shows a general architecture of ResNet50-SVM model.

4.3.2 MobileNetV2-SVM

Sandler et al. [185] proposed the MobileNetV2 architecture, presented in 2018 at the Computer Vision and Pattern Recognition (CVPR) conference, which improved the performance of mobile models. This model was intended to perform image classification and feature extraction tasks with a focus on efficiency and low weight, specifically optimized for mobile and embedded devices. Some of the most significant characteristics about MobileNetV2 are its inverted residuals, linear bottlenecks, expansion layer, width multiplier and resolution multiplier, skip connections, and global depthwise pooling.

MobileNetV2-SVM [186] was employed in this experiment. The final layer in MobileNetV2-50 was removed and substituted with an SVM to enhance the model's capacity to differentiate real and fake faces. L2 regularization was adopted as a form of lasso regression, and the loss function utilized was hinge loss. The overall structure of the MobileNetV2-SVM model is depicted in Fig. 4.5.

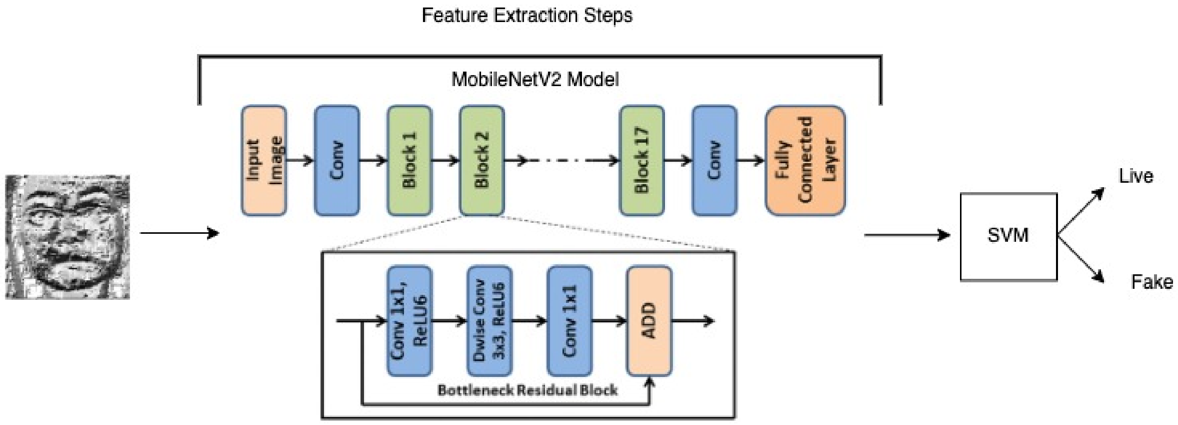


Figure 4.5: General architecture of MobileNetV2-SVM model.

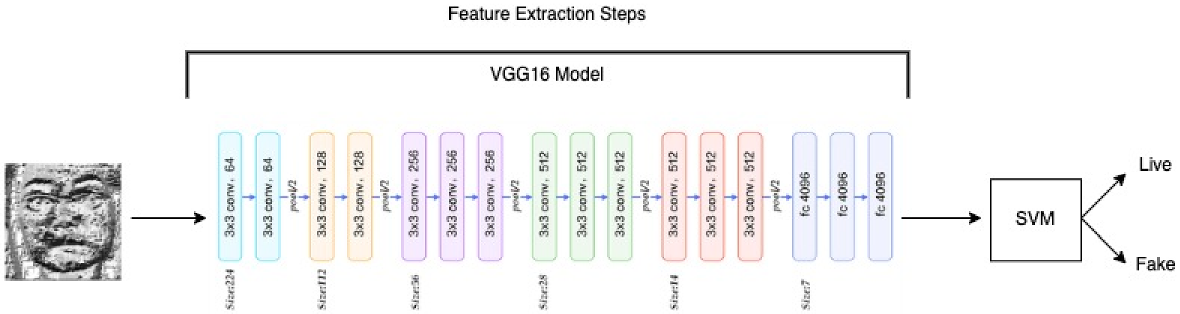


Figure 4.6: General architecture of VGG16-SVM model.

4.3.3 VGG16-SVM

Visual Geometry Group (VGG) at the University of Oxford introduced VGG16 [187]. VGG 16-layer denote its structure of 16 learnable layers, comprising 13 convolutional layers and 3 fully connected layers. The ImageNet dataset was utilized to train VGG16. This model showed the effectiveness of deep convolutional neural networks for image recognition tasks. Small convolution filters (3x3) and deep stacking are two of the main features of the VGG-16 architecture.

The experiment utilized VGG16-SVM [188]. In VGG16-50, the last layer was excluded and replaced with an SVM to improve the model's ability to distinguish between genuine and counterfeit faces. L2 regularization was applied as a lasso regression technique, and the chosen loss function was hinge loss. The general configuration of the VGG16-SVM model is illustrated in Fig. 4.6.

4.4 Proposed FedFacial Algorithm

Bakopoulou et al. [93] introduced SVM in Federated Learning setting and proposed a solution for mobile packet classification. Their work served as an inspiration for this piece. We proposed a hybrid CNN-SVM based model in Federated Learning which we used to classify real and fake faces. The Federated CNN-SVM technique is introduced in the Algorithm 2 presentation, whereby we utilize the hybrid CNN-SVM-based gradient updates in the Federated Averaging.

Algorithm 1 Proposed **FedFacial** model for PAD [189] [93] [190] [191]

```
Input: Given $K$ clients (indexed by $k$); $B$ local minibatch size; $L$ number of local epochs; $R$ number of global rounds; $C$ fraction of clients; $n_k$ is the training data size of client $k$; $n$ is the total data size from all users, $X$ is total number of images, $x$ is image and $\eta$ is learning rate.
Output: Using (LBP or Gabor filter) and Hybrid CNN-SVM based model for facial presentation attack detection
Global model learning (Server executes):
Initialize $\omega_0$ //(S1)
for each round $t = 1, 2, ..., R$ do
    $m \leftarrow max(C.K, 1)$
    $S_t \leftarrow$ (random set of $m$ clients)
    for each client $= 1, 2, ...K$ in parallel do
        $\omega_{t+1}^{k} \leftarrow$ ClientUpdate ( k, $\omega_t$) //(S4)
        $m_t \leftarrow \sum_{\in S_t} n_k$
        $\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{m_t} \omega_{t+1}^{t}$
Local model learning (Client Update($k$, $\omega$)):
Compute LBP features from $x$ and get feature matrix, $F$
$B_k \leftarrow$ (split of local $x$ image data into batches of size $B$)
for each local epoch $i = 1, 2, ..., L$ do // (S5)
    for batch $b \in$B$_k$ do
        $\omega \leftarrow \omega - \frac{\eta}{B} \sum_{i \in B_k} y_i.x_i$,when $y_i(\omega_i x_i) < 1$ // perform classification using CNN-SVM classifier (CT)
return $w$ to server (S2)
```

Details of model initialization, local model training, and global aggregation for Federated Learning are described in Section 2.15. A model summary of FedFacial is presented in Fig. 4.7

4.5 Performance Evaluation

Accuracy is a popular way to measure how well a machine learning model works. It measures how many of the model's guesses have been right out of all the predictions it has made. In

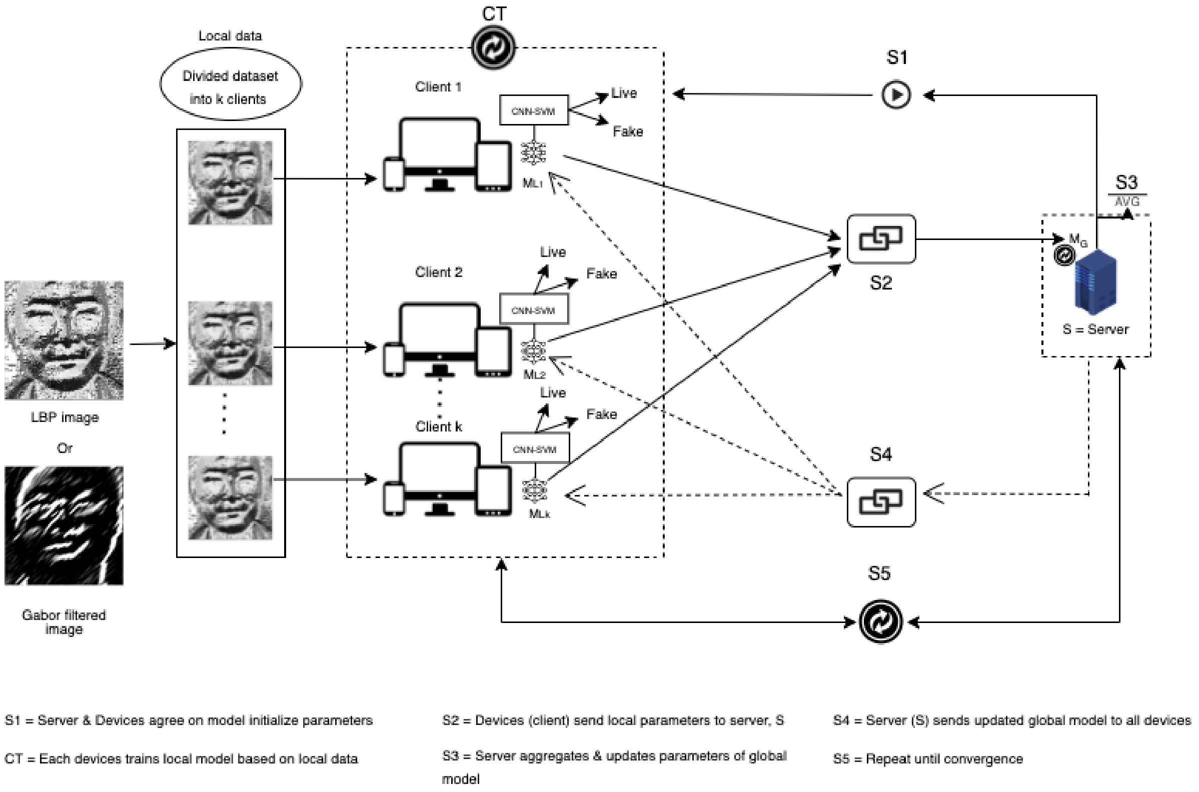

CT
Local data
Divided dataset into k clients
Client 1
CNN-SVM
Live
Fake
ML1
Client 2
Client k
MLk
ML2
LBP image
Or
Gabor filtered image
S1
S2
S3
AVG
MG
S = Server
S4
S5
S1 = Server & Devices agree on model initialize parameters
CT = Each devices trains local model based on local data
S2 = Devices (client) send local parameters to server, S
S3 = Server aggregates & updates parameters of global model
S4 = Server (S) sends updated global model to all devices
S5 = Repeat until convergence

Figure 4.7: Training process of our proposed FedFacial

other words, accuracy reveals how effectively the model classifies occurrences in the dataset on which it was tested. In a mathematical context, accuracy is derived through the following calculation:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \text{ x } 100\% \tag{4.1}$$

In this experiment, we used 2 clients. We applied 2 epochs for every client as a local round and 50 epochs as a global round at the server end. We analyzed validation accuracy for client 1 and client 2 by using the ResNet50, MobileNetV2, and VGG16 models. From the accuracy matrix, we notice that ResNet50 gives better results than MobileNetV2 and VGG16. However, we applied ResNet50-SVM, MobileNetV2-SVM, and VGG16-SVM models. Finally, we propose ResNet50-SVM in federated learning for facial presentation attack detection.

In this experiment, we use 2 clients. We apply 2 epochs for every client as a local round and 50 epochs as a global round. We analyze validation accuracy for client 1 and client 2 from ResNet50, MobileNetV2, and VGG16. From the accuracy matrix, we notice that ResNet50

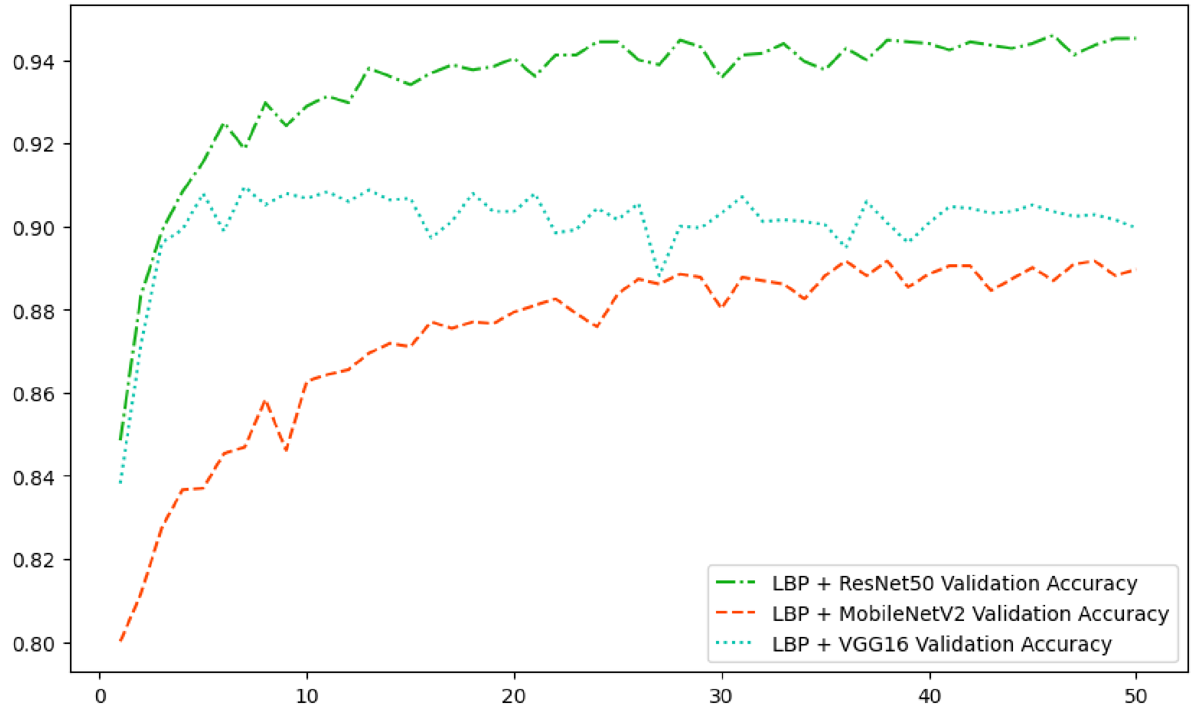


Figure 4.8: Validation accuracy (with LBP feature extraction method and CNNs)

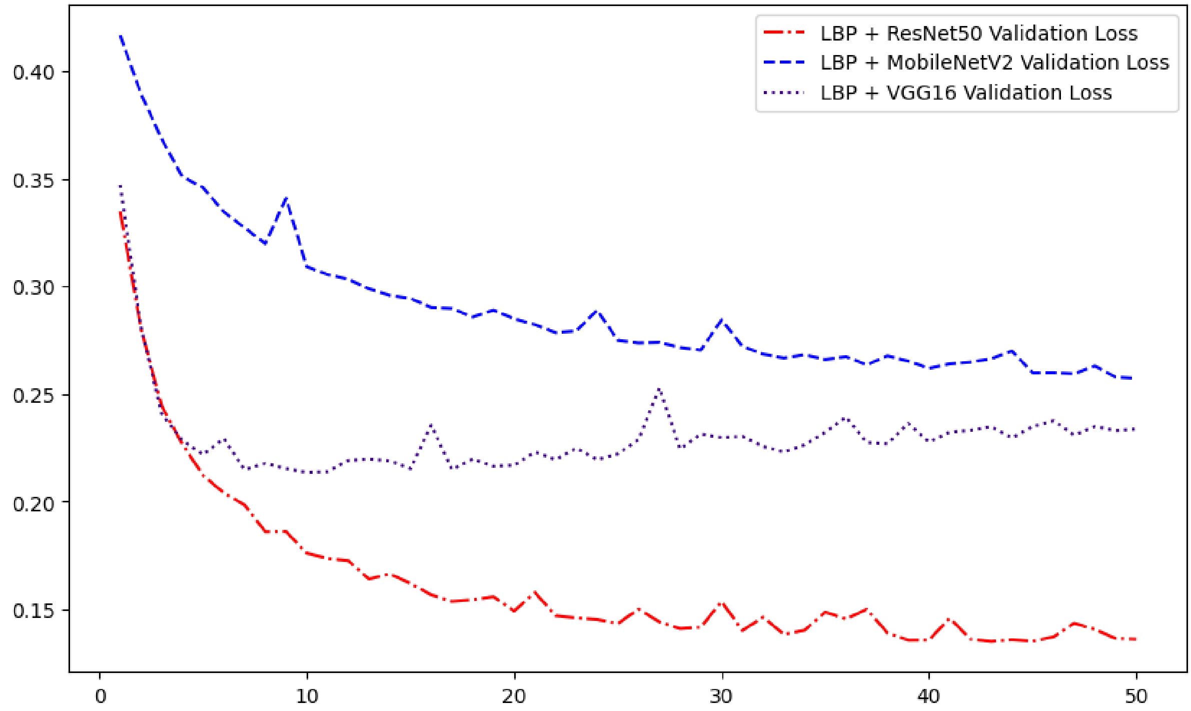


Figure 4.9: Validation loss (with LBP feature extraction method and CNNs)

Table 4.2: Validation loss and accuracy with applying the LBP + ResNet50 architecture

Global rounds	10	20	30	40	50
Client 1					
Loss	18.41%	15.63%	15.93%	14.02%	13.98%
Accuracy	92.39%	93.89%	92.78%	94.21%	94.29%
Client 2					
Loss	16.80%	14.22%	14.80%	13.14%	13.26%
Accuracy	93.41%	94.21%	94.37%	94.60%	94.76%

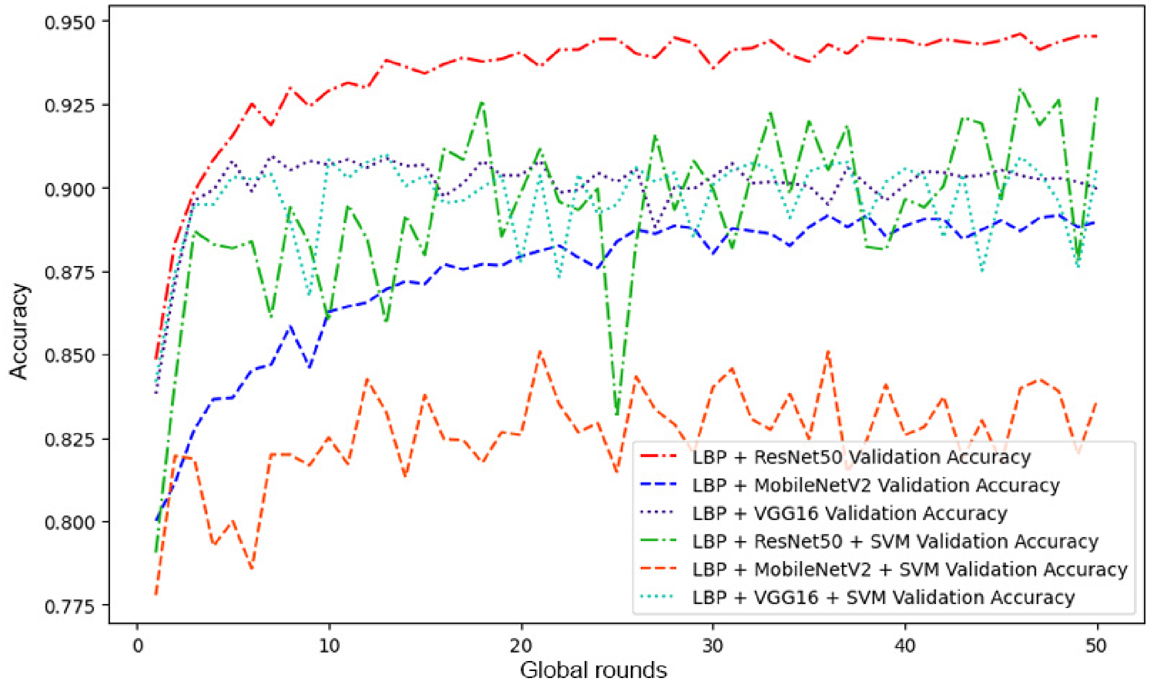


Figure 4.10: Validation accuracy (with LBP feature extraction method, CNNs and SVM)

Table 4.3: Validation loss and accuracy with applying the LBP + ResNet50-SVM architecture

Global rounds	10	20	30	40	50
Client 1					
Loss	29.37%	25.46%	25.18%	25.17%	26.18%
Accuracy	85.33%	89.06%	89.29%	88.90%	92.31%
Client 2					
Loss	27.14%	22.87%	22.72%	22.80%	24.18%
Accuracy	86.67%	90.63%	90.63%	90.40%	93.10%

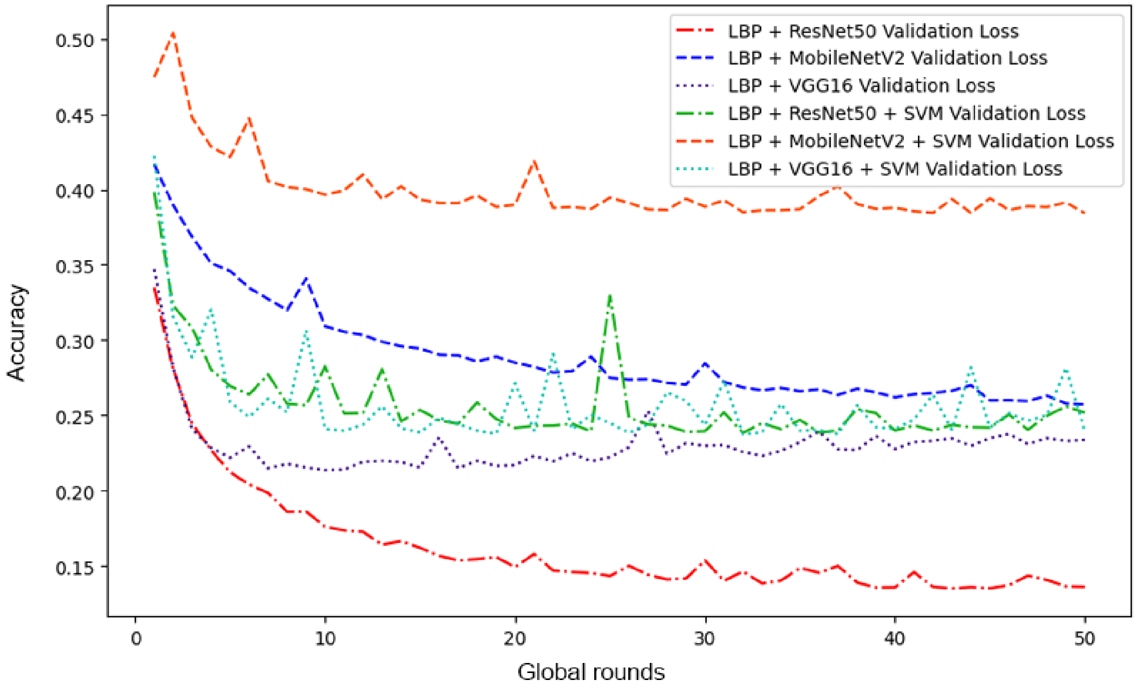


Figure 4.11: Validation loss (with LBP feature extraction method, CNNs and SVM)

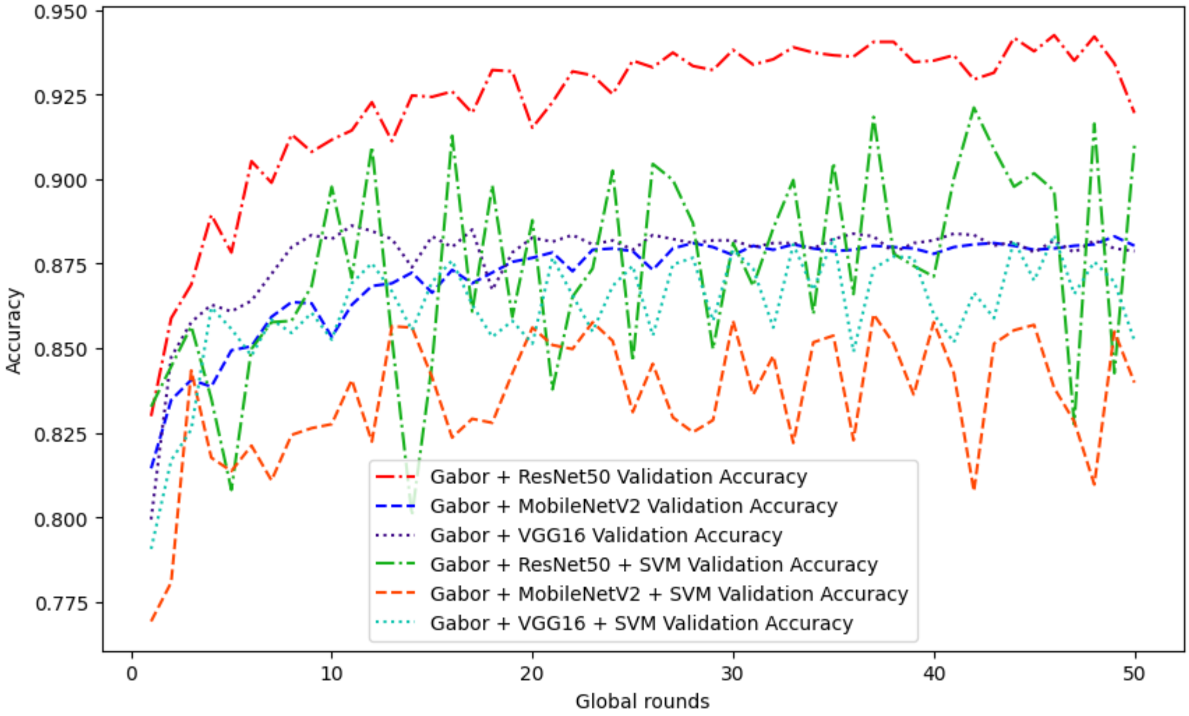


Figure 4.12: Validation accuracy (with LBP feature extraction method, CNNs and SVM)

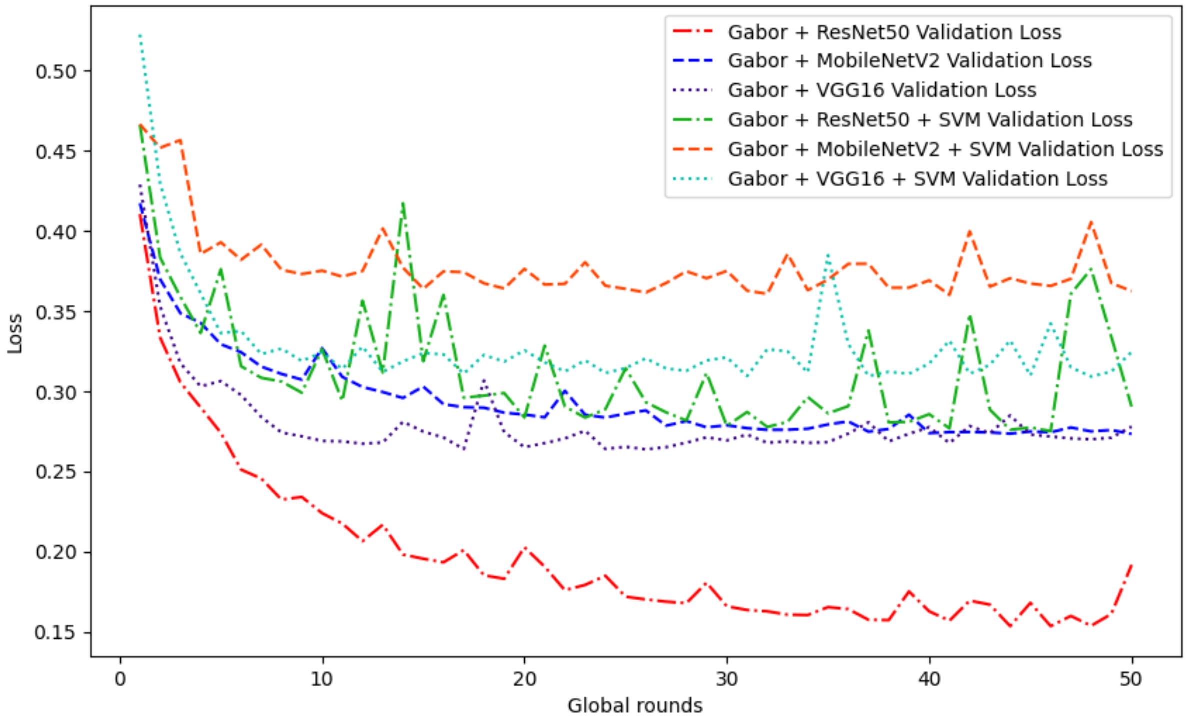


Figure 4.13: Validation loss (with LBP feature extraction method, CNNs and SVM)

Table 4.4: SOTA accuracy using NUAA dataset

Method	Accuracy
DoG filtered based [192]	87.5%
Using dynamic texture [193]	81.8%
Using fourier spectra [194]	84.5%
LTP [195]	91.1%
DLTP [195]	94.5%
ELBP [196]	95.1%
ResNet50 [197]	95.85%

gives better results than MobileNetV2 and VGG16. We receive the same result using SVM as the last layer. Finally, we propose ResNet50-SVM in federated learning for facial presentation attack detection.

4.6 Benchmarking Against the State-of-the-art Methods

Table 4.4 presents state-of-the-art methods accuracy and table 4.5 presents state-of-the-art methods for federated learning setting for facial Presentation Attack Detection problem.

Table 4.5: Facial presentation attack detection in federated learning

Method	Accuracy	HTER
Shao et al. (2020)	-	30.51%
Shao et al. (2021)	-	16.97%
Liu et al. (2022)	-	28.19%
LBP + ResNet50 + FL (Proposed)	94.53% (Avg)	-
LBP + ResNet50 + SVM + FL (Proposed)	92.71% (Avg)	-
Gabor + ResNet50 + FL (Proposed)	93.82% (Avg)	-
Gabor + ResNet50 + SVM + FL (Proposed)	91% (Avg)	-

CHAPTER V

FINGERPRINT PRESENTATION ATTACK DETECTION

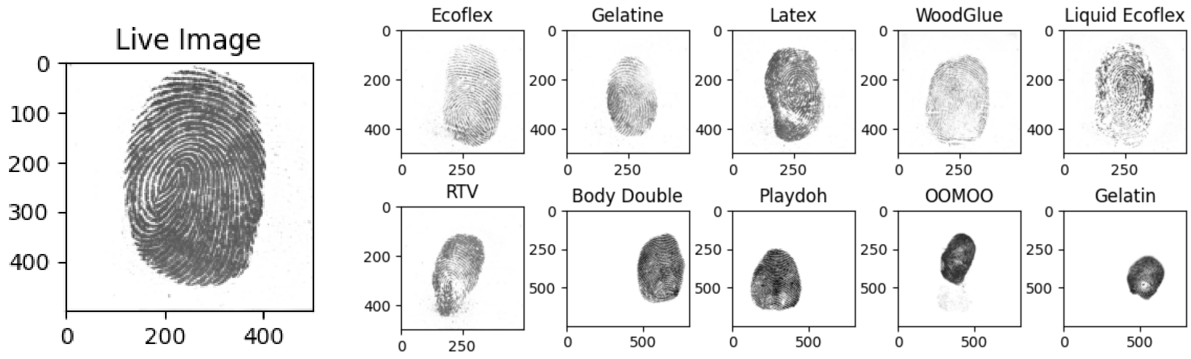
5.1 Dataset

We used Fingerprint Liveness Detection Competition (LivDet) 2015 dataset [198] for our experiment. This dataset is made from four different fingerprint scanning machines, such as Green Bit (model name: DactyScan26), Biometrika (model name: HiScan-PRO), Digital Persona (model name: U.are.U 5160), and RTV (model name: L Scan Guardian). For the Green Bit, the Biometrika, and the Digital Persona datasets, Ecoflex, gelatine, latex, woodglue, a liquid Ecoflex, and RTV (a two-component silicone rubber) are adopted as spoofing materials. For the Crossmatch dataset, Playdoh, Body Double, Ecoflex, OOMOO (a silicone rubber), and a novel form of Gelatin are used as spoofing materials. The whole dataset is divided into two sets, such as training and testing.

For the Green Bit, the Biometrika, and the Digital Persona datasets, every dataset has 1000 live images, and every dataset has 250 images for ecoflex, gelatine, latex, woodglue, liquid ecoflex, and RTV. For the crossmatch dataset, it has 1000 live images, 300 body double images, 270 ecoflex images, 281 playdoh images, 297 OOMOO images, and 300 gelatin images. Table 5.1 presents a whole summary for LivDet 2015 dataset. Fig. 5.1 presents one live fingerprint image and other spoofed or fake fingerprint images.

5.2 Feature extraction

Local Binary Pattern: LBP is one of the most popular feature extraction techniques to detect fake fingerprints. We already discussed LBP for facial Presentation Attack Detection in Sections 2.6.3.1 and 4.2.



(a) Live image

(b) Spoof images by different materials

Figure 5.1: Sample images from LivDet dataset

Table 5.1: A summary of the dataset used in this study (LivDet 2015)

Fingerprint Reader	Green Bit	Biometrika	Digital Persona	Crossmatch
Device Model	DactyScan26	HiScan-PRO	U.are.U 5160	L Scan Guardian
Image Size Resolution (dpi)	500 x 500 500	1000 x 1000 1000	252 x 324 500	640 x 480 500
#Live Images Train / Test	1000 / 1000	1000 / 1000	1000 / 1000	1510 / 1500
#Spoof Images Train / Test	1000 / 1500	1000 / 1500	1000 / 1500	1473 / 1448
Spoof Materials	Ecoflex, Gelatine, Latex, Wood Glue Liquid Ecoflex, RTV			Body Double PlayDoh, OOMOO Ecoflex, Gelatin

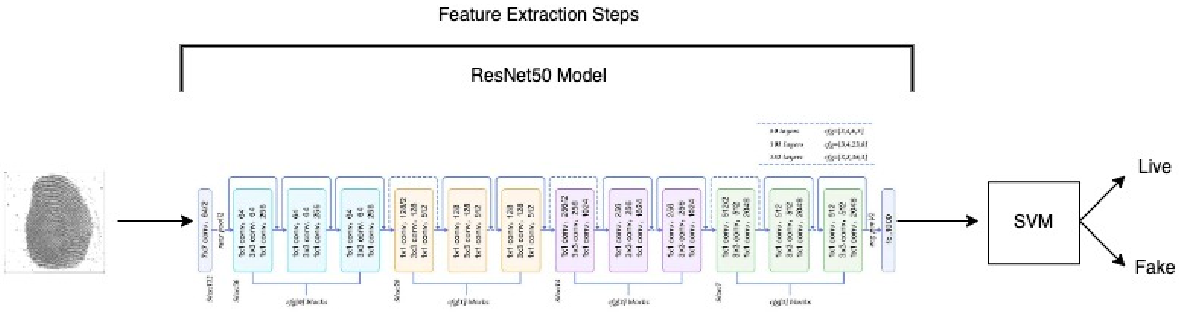


Figure 5.2: General architecture of ResNet-SVM model.

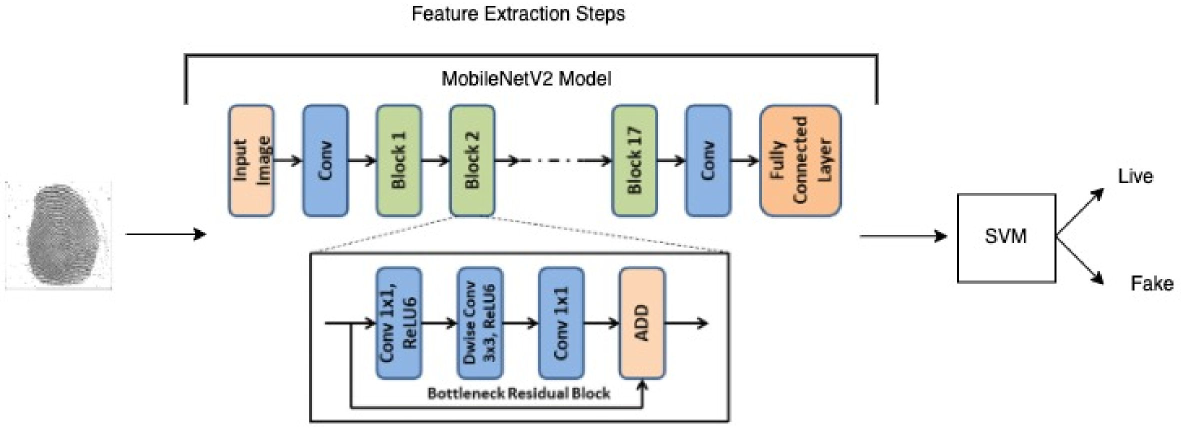


Figure 5.3: General architecture of MobileNetV2-SVM model.

5.3 Models

As hybrid CNN-SVM model, we used transfer learning based approaches such as ResNet-SVM, MobileNetV2-SVM, VGG16-SVM. Section 4.3.1 describes details of ResNet-SVM, Section 4.3.2 describes details of MobileNetV2-SVM, and Section 5.3 describes details of VGG16-SVM. Also, Figs. 5.2, 5.3, and 5.4 present general architecture, respectively, for ResNet-SVM, MobileNetV2-SVM, and VGG16-SVM.

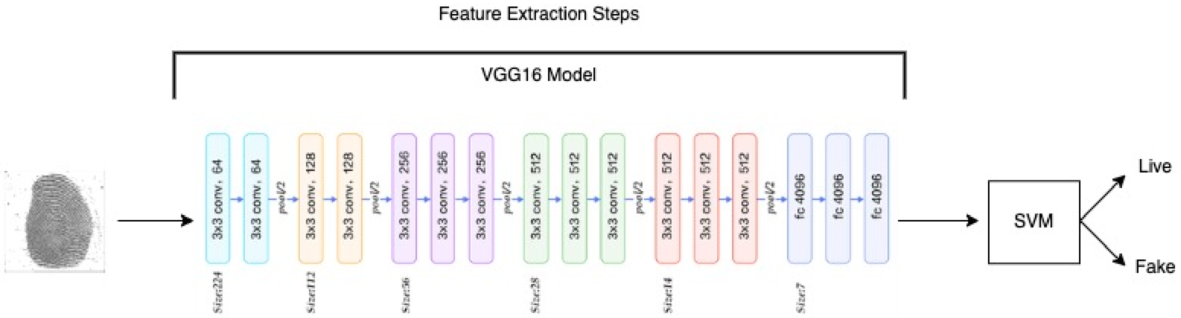


Figure 5.4: General architecture of VGG16-SVM model.

5.4 Proposed FedThumb Algorithm

Bakopoulou et al. [93] introduced SVM in Federated Learning setting and proposed a solution for mobile packet classification. Their work served as an inspiration for this piece. We proposed a hybrid CNN-SVM based model in Federated Learning which we used to classify real and fake fingerprints. The Federated CNN-SVM technique is introduced in the Algorithm 2 presentation, whereby we utilize the hybrid CNN-SVM-based gradient updates in the Federated Averaging.

Algorithm 2 Proposed **FedThumb** model for PAD [189] [93] [190] [191]

```
Input: Given $K$ clients (indexed by $k$); $B$ local minibatch size; $L$ number of local epochs; $R$ number of global rounds; $C$ fraction of clients; $n_k$ is the training data size of client $k$; $n$ is the total data size from all users, $X$ is total number of images, $x$ is image and $\eta$ is learning rate.
Output: Using (LBP or HOG) and Hybrid CNN-SVM based model for facial presentation attack detection
Global model learning (Server executes):
Initialize $\omega_0$ //(S1)
for each round $t = 1, 2, ..., R$ do
    $m \leftarrow max(C.K, 1)$
    $S_t \leftarrow$ (random set of $m$ clients)
    for each client $= 1, 2, ...K$ in parallel do
        $\omega_{t+1}^{k} \leftarrow$ ClientUpdate ( k, $\omega_t$) //(S4)
        $m_t \leftarrow \sum_{\in S_t} n_k$
        $\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{m_t} \omega_{t+1}^{t}$
Local model learning (Client Update($k, \omega$)):
Compute LBP features from $x$ and get feature matrix, $F$
$B_k \leftarrow$ (split of local $x$ image data into batches of size $B$)
for each local epoch $i = 1, 2, ..., L$ do // (S5)
    for batch $b \in \mathrm{B}_k$ do
        $\omega \leftarrow \omega - \frac{\eta}{B} \sum_{i \in B_k} y_i.x_i$,when $y_i(\omega_i x_i) < 1$ // perform classification using CNN-SVM classifier (CT)
return $w$ $to$ $server$ $(S2)$
```

Details of model initialization, local model training, and global aggregation for Federated Learning are described in Section 2.15. A model summary of FedThumb is presented in Fig. 5.5

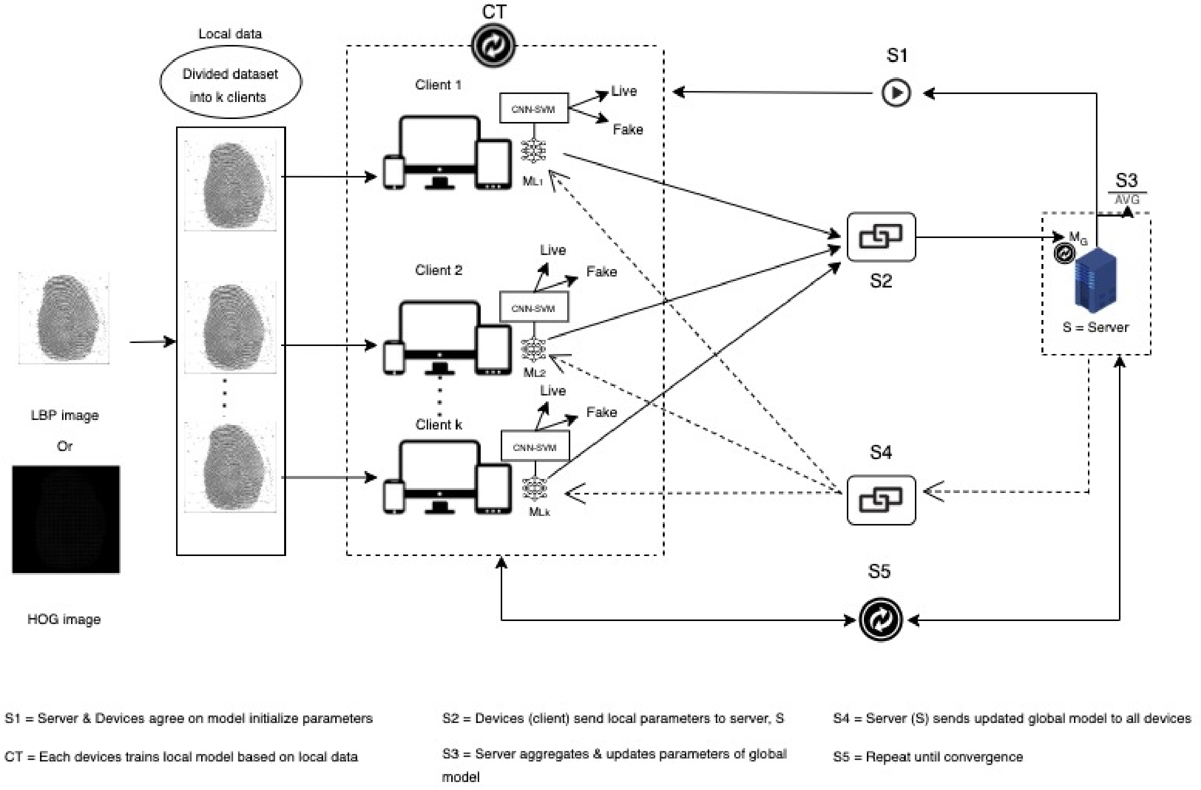

CT
Local data
Divided dataset into k clients
Client 1
CNN-SVM
Live
Fake
ML1
Client 2
CNN-SVM
Live
Fake
ML2
Client k
CNN-SVM
Live
Fake
MLk
LBP image
Or
HOG image
S1
S2
S3
AVG
MG
S = Server
S4
S5
S1 = Server & Devices agree on model initialize parameters
CT = Each devices trains local model based on local data
S2 = Devices (client) send local parameters to server, S
S3 = Server aggregates & updates parameters of global model
S4 = Server (S) sends updated global model to all devices
S5 = Repeat until convergence

Figure 5.5: Training process of our proposed FedFacial

5.5 Performance Evaluation

Accuracy is a popular way to measure how well a machine learning model works. It measures how many of the model's guesses have been right out of all the predictions it has made. In other words, accuracy reveals how effectively the model classifies occurrences in the dataset on which it was tested. In a mathematical context, accuracy is derived through the following calculation:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \text{ x } 100\% \tag{5.1}$$

In this experiment, we used 2 clients. We applied 2 epochs for every client as a local round and 50 epochs as a global round at the server end. We analyzed validation accuracy for client 1 and client 2 by using the CNN, ResNet50, MobileNetV2, and VGG16 models. From the accuracy matrix, we notice that CNN gives better results than ResNet50, MobileNetV2 and VGG16. However, we applied ResNet50-SVM, MobileNetV2-SVM, and VGG16-SVM

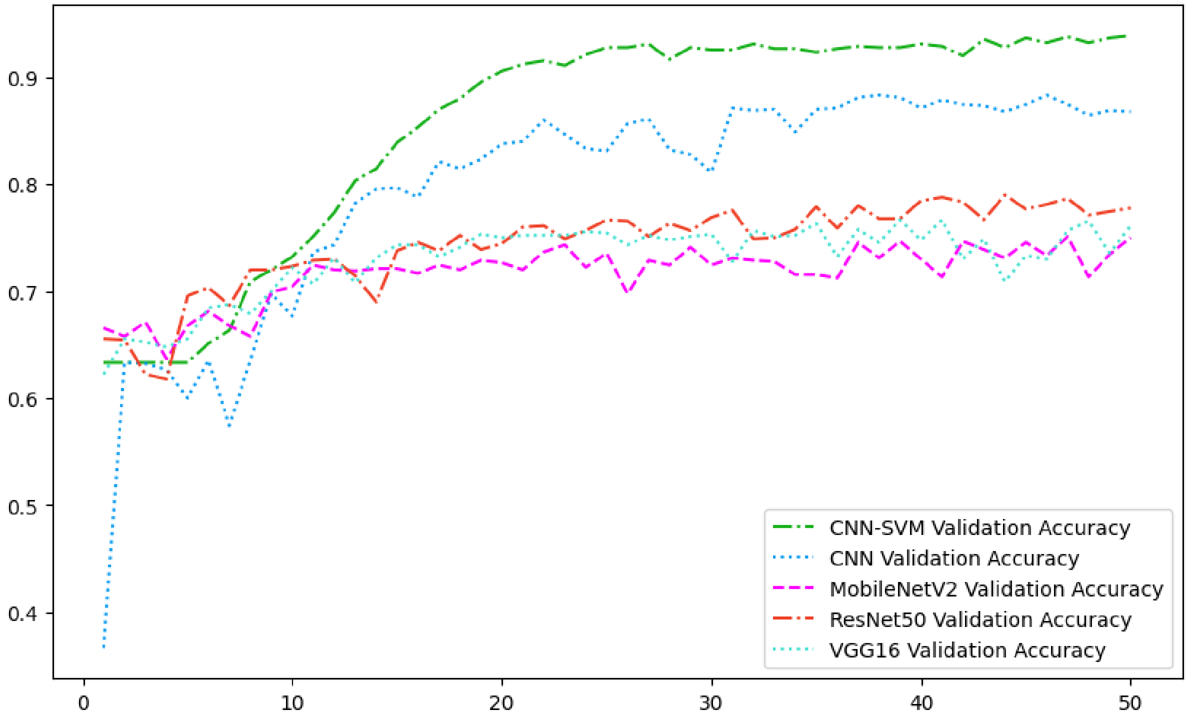


Figure 5.6: Validation accuracy

Table 5.2: Fingerprint presentation attack detection in federated learning

Method	Evaluation Metrics
CNN-SVM (Proposed)	93.89% (Accuracy)
Zhang et al.(2019)	95.25% (Accuracy)
Abdullahi et al. (2022)	0.31%(LivDet 2013), 2.26%(LivDet 2015) [ACE]
Saguy et al.(2021)	21%[FRR]
Pałka et al.(2020)	87.9%[TDR], 3.9%[FDR] (Tim Frequency)

models. Finally, we propose CNN-SVM in federated learning for fingerprint presentation attack detection.

5.6 Benchmarking Against the State-of-the-art Methods

Table 5.2 presents state-of-the-art methods accuracy for fingerprint Presentation Attack Detection problem.

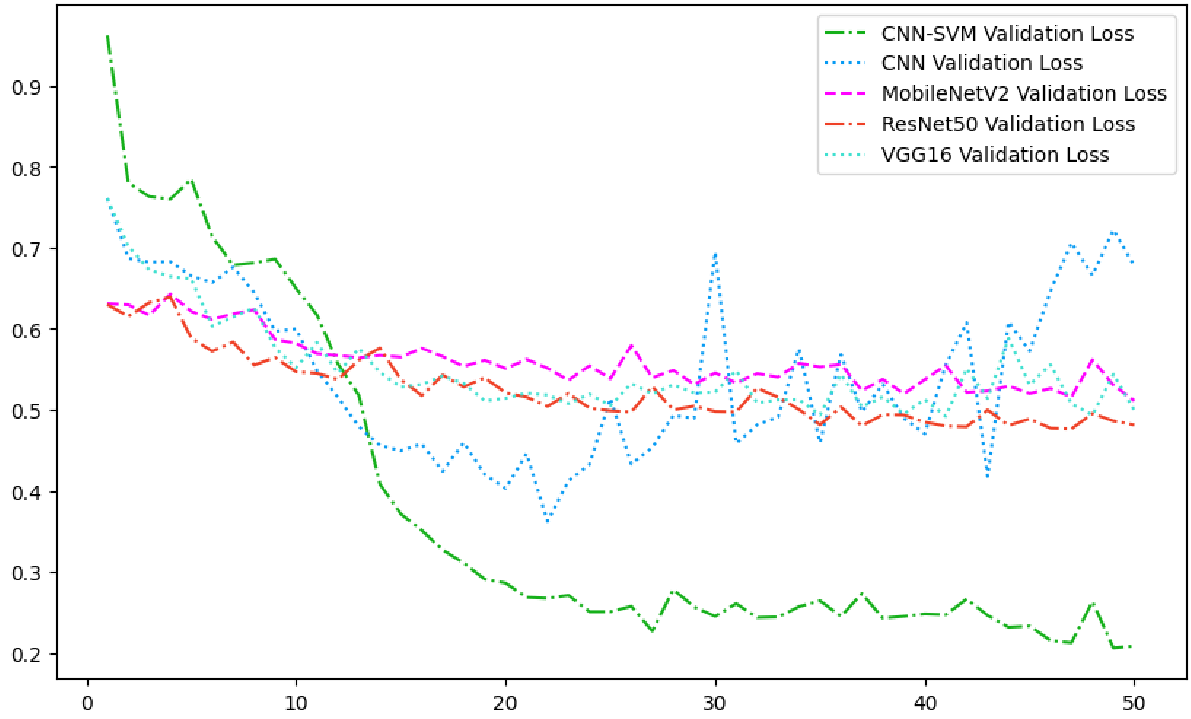


Figure 5.7: Validation loss

CHAPTER VI

CONCLUDING REMARKS

6.1 Summary

6.1.1 Facial presentation attack detection

In this study, we proposed two federated learning-based methods: Local binary pattern (LBP) or gabor filter and ResNet50 based; another one is Local binary pattern (LBP) or gabor filter, ResNet50 and SVM based. We calculated LBP and gabor filter images and then classified those images by hybrid CNN-SVM architectures such as ResNet50, MobileNetV2, and VGG16 in federated learning setting. Experiment findings show that the proposed federated learning model performs similarly to state-of-the-art centralized machine learning setting.

6.1.2 Fingerprint presentation attack detection

In this study, we proposed two combined descriptors to extract image features. Local binary pattern (LBP), which captures texture patterns and local variations, and histogram of oriented gradient (HOG), which captures shape and edge information. We calculated LBP and HOG images and then classified those images by support vector machine (SVM) in federated learning setting. Experiment findings show that the proposed federated learning model is robust to common spoofing materials, including ecoflex, gelatine, latex, wood glue, liquid ecoflex, RTV, body double, playdoh, OOMOO, and gelatin.

In order to investigate the liveness attributes in detail, our next study will need to expand the method with other feature descriptors of fingerprint images, such as speeded-up robust feature to check scale and rotation changes, local phase pattern to analyze captured local phase information, and gabor filters or circular gabor filter-based features to analyze local frequency and orientation content. Additionally, we need to explore shape features to analyze minutiae points, pore distribution, and ridge curvature. Also, we need to find out if the proposed way can help boost the accuracy and time complexity of our model by using different federated learning algorithms.

6.2 Future directions

6.2.1 Facial Presentation Attack Detection

- To our best knowledge, this paper proposes ResNet50-SVM in federated learning setting for the first time.
- We evaluated the accuracy matrix of various CNN-SVM models like MobilenetV2-SVM and VGG16-SVM.
- We compared our work to other state-of-the-art methods and observed that, in distributed machine learning settings like FL, it had higher or similar validation accuracy.

In order to investigate the PAD attributes in detail, our next study will need to expand the method with other CNN based architectures and fine-tune our model to get a better result. Also, we would like to explore privacy costs for PAD.

6.2.2 Fingerprint Presentation Attack Detection

In this study, we proposed two combined descriptors to extract image features. Local binary pattern (LBP), which captures texture patterns and local variations, and histogram of oriented gradient (HOG), which captures shape and edge information. We calculated LBP and HOG images and then classified those images by support vector machine (SVM) in federated learning setting. Experiment findings show that the proposed federated learning model is robust to common spoofing materials, including ecoflex, gelatine, latex, wood glue, liquid ecoflex, RTV, body double, playdoh, OOMOO, and gelatin.

In order to investigate the liveness attributes in detail, our next study will need to expand the method with other feature descriptors of fingerprint images, such as speeded-up robust feature to check scale and rotation changes, local phase pattern to analyze captured local phase information, and gabor filters or circular gabor filter-based features to analyze local frequency and orientation content. Additionally, we need to explore shape features to analyze minutiae points, pore distribution, and ridge curvature. Also, we need to find out if the proposed way can help boost the accuracy and time complexity of our model by using different federated learning algorithms.

6.3 Funding acknowledgements

The University of Texas Rio Grande Valley provided support to the first author, S M Sarwar, through Presidential Research Fellowship Awards. The experiment run on UTRGV HPC is supported by NSF grant number 2018900 and DoD grant number W911NF2110169.

REFERENCES

[1] ISO/IEC JTC 1/SC 37 Biometrics, "Information technology — biometric presentation attack detection — part 1: Framework," *35.240.15 Identification cards. Chip cards. Biometrics*, Jan. 2016, https://www.iso.org/standard/53227.html.

[2] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of biometrics*. Springer Science & Business Media, 2007.

[3] K. H. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *The Journal of the Acoustical Society of America*, vol. 24, no. 6, pp. 637–642, 1952.

[4] W. W. Bledsoe, "The model method in facial recognition," *Panoramic Research Inc., Palo Alto, CA, Rep. PR1*, vol. 15, no. 47, p. 2, 1966.

[5] M. D. Kelly, *Visual identification of people by computer*. Stanford University, 1971.

[6] Statista Research Department, "Types of biometric technologies used in applications in the u.s. 2018," *Statista*, Jul. 2023, accessed: 2023-07-30. [Online]. Available: www.statista.com/statistics/934322/united-states-biometric-technology-types-use/

[7] "Aadhaar-based payments widely used but remain vulnerable to frauds," *ETV Bharat*, Jul. 2023, accessed: 2023-07-31. [Online]. Available: www.etvbharat.com/english/bharat/aadhaar-based-payments-widely-used-but-remain-vulnerable-to-frauds/na20230725214046729729226

[8] H. Dewey-Hagborg, "Stranger visions: A provocation," *IEEE Security Privacy*, vol. 11, no. 6, pp. 69–70, 2013, doi: 10.1109/MSP.2013.152.

[9] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: A public database and a baseline," in *2011 International Joint Conference on Biometrics (IJCB)*, 2011, pp. 1–7, 10.1109/IJCB.2011.6117503.

[10] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI 11*. Springer, 2010, pp. 504–517.

[11] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *2012 BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG)*. IEEE, 2012, pp. 1–7.

[12] N. Erdogmus and S. Marcel, "Spoofing face recognition with 3d masks," *IEEE transactions on information forensics and security*, vol. 9, no. 7, pp. 1084–1097, 2014.

[13] I. Chingovska, N. Erdogmus, A. Anjos, and S. Marcel, “Face recognition systems under spoofing attacks,” *Face Recognition Across the Imaging Spectrum*, pp. 165–194, 2016.

[14] J. Galbally, S. Marcel, and J. Fierrez, “Biometric antispoofing methods: A survey in face recognition,” *IEEE Access*, vol. 2, pp. 1530–1552, 2014.

[15] “The man in the latex mask: Black serial armed robber disguised himself as a white man to rob betting shops,” *Daily Mail*, Jun. 2012, www.dailymail.co.uk/news/article-2153346/Black-armed-robber-disguised-white-man-using-latex-mask.html.

[16] N. M. Duc and B. Q. Minh, “Your face is not your password face authentication bypassing lenovo–asus–toshiba,” *Black Hat Briefings*, vol. 4, p. 158, 2009.

[17] “Smartphone face recognition bypassed with 2d photo, research finds,” *Yahoo Finance UK*, May 2023, accessed: 2023-07-31. [Online]. Available: www.uk.finance.yahoo.com/news/smartphone-face-recognition-bypassed-with-2d-photo-research-finds-230140964.html

[18] S.-Q. Liu and P. C. Yuen, “Recent progress on face presentation attack detection of 3d mask attack,” *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection and Vulnerability Assessment*, pp. 231–259, 2023.

[19] A. Agarwal, D. Yadav, N. Kohli, R. Singh, M. Vatsa, and A. Noore, “Face presentation attack with latex masks in multispectral videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 81–89.

[20] P. P. Chan and Y. Shu, “Face liveness detection by brightness difference,” in *Machine Learning and Cybernetics: 13th International Conference, Lanzhou, China, July 13-16, 2014. Proceedings 13.* Springer, 2014, pp. 144–150.

[21] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, “Oulu-npu: A mobile face presentation attack database with real-world variations,” in *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017).* IEEE, 2017, pp. 612–618.

[22] A. George and S. Marcel, “Multi-channel face presentation attack detection using deep learning,” *Deep Learning-Based Face Analytics*, pp. 269–304, 2021.

[23] D. Sharma and A. Selwal, “A survey on face presentation attack detection mechanisms: hitherto and future perspectives,” *Multimedia Systems*, vol. 29, no. 3, pp. 1527–1577, 2023.

[24] R. Raghavendra, K. B. Raja, and C. Busch, “Presentation attack detection for face recognition using light field camera,” *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1060–1075, 2015.

[25] L. Sun, G. Pan, Z. Wu, and S. Lao, "Blinking-based live face detection using conditional random fields," in *Advances in Biometrics: International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007. Proceedings.* Springer, 2007, pp. 252–260.

[26] A. Lagorio, M. Tistarelli, M. Cadoni, C. Fookes, and S. Sridharan, "Liveness detection based on 3d face shape analysis," in *2013 International Workshop on Biometrics and Forensics (IWBF).* IEEE, 2013, pp. 1–4.

[27] M.-A. Waris, H. Zhang, I. Ahmad, S. Kiranyaz, and M. Gabbouj, "Analysis of textural features for face biometric anti-spoofing," in *21st European Signal Processing Conference (EUSIPCO 2013).* IEEE, 2013, pp. 1–5.

[28] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *2011 international joint conference on Biometrics (IJCB).* IEEE, 2011, pp. 1–7.

[29] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *2012 5th IAPR international conference on Biometrics (ICB).* IEEE, 2012, pp. 26–31.

[30] "ios 15 includes improved face id anti-spoofing models and other vulnerability fixes," *MacRumors*, Sep. 2021, accessed: 2023-07-31. [Online]. Available: www.macrumors.com/2021/09/20/ios-15-improved-face-id-anti-spoofing/

[31] "In apple's shadow, google takes new route to face recognition on pixel phones," *Reuters*, Oct. 2022, accessed: 2023-08-01. [Online]. Available: www.reuters.com/technology/apples-shadow-google-takes-new-route-face-recognition-pixel-phones-2022-10-06

[32] "Get your german interior minister's fingerprint here," *The Register*, Mar. 2008, accessed: 2023-07-30. [Online]. Available: www.theregister.com/2008/03/30/german_interior_minister_fingerprint_appropriated/

[33] "iphone 5s fingerprint sensor hacked by germany's chaos computer club," *The Guardian*, Sep. 2013, accessed: 2023-07-30. [Online]. Available: www.theguardian.com/technology/2013/sep/22/apple-iphone-fingerprint-scanner-hacked

[34] "Malaysia car thieves steal finger," *BBC*, Mar. 2005, accessed: 2023-07-30. [Online]. Available: www.news.bbc.co.uk/2/hi/asia-pacific/4396831.stm

[35] "Fake fingers fool hospital clock-in scanner," *Sky News*, Mar. 2013, accessed: 2023-07-30. [Online]. Available: www.news.sky.com/story/fake-fingers-fool-hospital-clock-in-scanner-10451918

[36] "Exclusive: Man in disguise boards international flight," *CNN*, Nov. 2010, accessed: 2023-07-30. [Online]. Available: www.cnn.com/2010/WORLD/americas/11/04/canada.disguised.passenger/index.html

[37] S. Jia, G. Guo, and Z. Xu, "A survey on 3d mask presentation attack detection and countermeasures," *Pattern recognition*, vol. 98, p. 107032, 2020.

[38] "Fingerprint lock in samsung galaxy 5 easily defeated by whitehat hackers," *Arstechnica*, Apr. 2014, accessed: 2023-07-31. [Online]. Available: www.arstechnica.com/information-technology/2014/04/fingerprint-lock-in-samsung-galaxy-5-easily-defeated-by-whitehat-hackers/

[39] "Samsung and huawei fingerprint scanners can be fooled using an inkjet printer," *The Guardian*, Mar. 2016, accessed: 2023-07-31. [Online]. Available: www.theguardian.com/technology/2016/mar/08/samsung-and-huawei-fingerprint-scanners-can-be-fooled-using-an-inkjet-printer

[40] P. Bontrager, A. Roy, J. Togelius, N. Memon, and A. Ross, "Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS).* IEEE, 2018, pp. 1–9.

[41] P. Coli, G. L. Marcialis, and F. Roli, "Fingerprint silicon replicas: static and dynamic features for vitality detection using an optical capture device," *International Journal of Image and Graphics*, vol. 8, no. 04, pp. 495–512, 2008, doi:10.1142/S0219467808003209.

[42] L. N. Darlow, J. Connan, and S. S. Akhoury, "Internal fingerprint zone detection in optical coherence tomography fingertip scans," *Journal of Electronic Imaging*, vol. 24, no. 2, pp. 023 027–023 027, 2015.

[43] I. Goicoechea-Telleria, K. Kiyokawa, J. Liu-Jimenez, and R. Sanchez-Reillo, "Low-cost and efficient hardware solution for presentation attack detection in fingerprint biometrics using special lighting microscopes," *IEEE Access*, vol. 7, pp. 7184–7193, 2019.

[44] M. Hammad and K. Wang, "Parallel score fusion of ecg and fingerprint for human authentication based on convolution neural network," *Computers & Security*, vol. 81, pp. 107–122, 2019.

[45] P. Keilbach, J. Kolberg, M. Gomez-Barrero, C. Busch, and H. Langweg, "Fingerprint presentation attack detection using laser speckle contrast imaging," in *2018 international conference of the biometrics special interest group (BIOSIG).* IEEE, 2018, pp. 1–6.

[46] A. Abhyankar and S. Schuckers, "Fingerprint liveness detection using local ridge frequencies and multiresolution texture analysis techniques," in *2006 International Conference on Image Processing*, 2006, pp. 321–324, 10.1109/ICIP.2006.313158.

[47] J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia, "Fingerprint liveness detection based on quality measures," in *2009 First IEEE International Conference on Biometrics, Identity and Security (BIdS).* IEEE, 2009, pp. 1–8.

[48] C. Gottschlich, E. Marasco, A. Y. Yang, and B. Cukic, “Fingerprint liveness detection based on histograms of invariant gradients,” in *IEEE International Joint Conference on Biometrics*, 2014, pp. 1–7.

[49] I. Goicoechea-Telleria, K. Kiyokawa, J. Liu-Jimenez, and R. Sanchez-Reillo, “Low-cost and efficient hardware solution for presentation attack detection in fingerprint biometrics using special lighting microscopes,” *IEEE Access*, vol. 7, pp. 7184–7193, 2019.

[50] E. Marasco and A. Ross, “A survey on antispoofing schemes for fingerprint recognition systems,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, pp. 1–36, 2014.

[51] D. Ametefe, S. Sarnin, D. Ali, and M. Zaheer, “Fingerprint liveness detection schemes: A review on presentation attack,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 10, no. 2, pp. 217–240, 2022.

[52] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, “Digital image processing using matlab,” *Gatesmark Publishing*, 2009.

[53] J. D. Foley, A. V. Dam, and S. K. Feiner, “Introduction to computer graphics,” *Addison-Wesley Professional*, 1993.

[54] C. Poynton, “Digital video and hd: Algorithms and interfaces,” *Morgan Kaufmann*, 2012.

[55] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986, doi: 10.1109/TPAMI.1986.4767851.

[56] A. Fitzgibbon, M. Pilu, and R. Fisher, “Direct least square fitting of ellipses,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, 1999, doi: 10.1109/34.765658.

[57] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern Recognition*, vol. 29, 1996, doi: 10.1016/0031-3203(95)00067-4.

[58] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, pp. 886–893 vol. 1, doi: 10.1109/CVPR.2005.177.

[59] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *J. Opt. Soc. Am. A*, vol. 2, no. 7, pp. 1160–1169, 1985, doi: 10.1364/JOSAA.2.001160.

[60] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973, doi: 10.1109/TSMC.1973.4309314.

[61] M. M. Galloway, "Texture analysis using gray level run lengths," vol. 4, no. 2, pp. 172–179, 1975, doi: 10.1016/S0146-664X(75)80008-6.

[62] M. Chihaoui, A. Elkefi, W. Bellil, and C. Ben Amar, "A survey of 2d face recognition techniques," *Computers*, vol. 5, no. 4, 2016, doi: 10.3390/computers5040021.

[63] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 7, pp. 971–987, 2002, doi: 10.1109/TPAMI.2002.1017623.

[64] D. Gabor, "Theory of communication," *Institution of Electrical Engineers*, vol. 93, no. 3, pp. 429–457, 1946.

[65] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, no. 10, pp. 847–856, doi: 10.1016/0042-6989(80)90065-6.

[66] J. Yang, L. Liu, T. Jiang, and Y. Fan, "A modified gabor filter design method for fingerprint image enhancement," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1805–1817, doi: 10.1016/S0167-8655(03)00005-9.

[67] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1, doi: 10.1109/CVPR.2005.177.

[68] S. Zhang and X. Wang, "Human detection and object tracking based on histograms of oriented gradients," in *2013 Ninth International Conference on Natural Computation (ICNC)*, pp. 1349–1353, doi: 10.1109/ICNC.2013.6818189.

[69] "High performance computing," *University of Texas Rio Grande Valley HPC*, accessed: 2023-07-31. [Online]. Available: https://hpc.utrgv.edu/

[70] A. Jung, *Machine Learning: The Basics.* Springer, Singapore, 2022, https://alexjungaalto.github.io/MLBasicsBook.pdf.

[71] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia: case studies on organization and retrieval.* Springer, 2008, pp. 21–49.

[72] B. Hu and J. Shao, "Generalized linear model selection using r2," *Journal of statistical planning and inference*, vol. 138, no. 12, pp. 3705–3712, 2008.

[73] "What is a machine learning model?" MATHWORKS, accessed: 2023-07-31. [Online]. Available: www.mathworks.com/discovery/machine-learning.html

[74] Z. Ghahramani, "Unsupervised learning," in *Summer school on machine learning.* Springer, 2003, pp. 72–112.

[75] S. Chander and P. Vijaya, "Unsupervised learning methods for data clustering," in *Artificial Intelligence in Data Mining.* Elsevier, 2021, pp. 41–64.

[76] H. Valpola, “From neural pca to deep unsupervised learning,” in *Advances in independent component analysis and learning machines*. Elsevier, 2015, pp. 143–171.

[77] X. Zhu and A. B. Goldberg, *Introduction to semi-supervised learning*. Springer Nature, 2022.

[78] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[79] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[80] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.

[81] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[82] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[83] “What is a convolutional neural network?” MATHWORKS, accessed: 2023-08-2. [Online]. Available: www.mathworks.com/discovery/convolutional-neural-network-matlab.html

[84] C. Öztürk, M. Taşyürek, and M. U. Türkdamar, “Transfer learning and fine-tuned transfer learning methods’ effectiveness analyse in the cnn-based deep learning models,” *Concurrency and Computation: Practice and Experience*, vol. 35, no. 4, p. e7542, 2023.

[85] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[86] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[87] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[88] Y. Tang, “Deep learning using linear support vector machines,” *arXiv preprint arXiv:1306.0239*, 2013.

[89] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” *Advances in neural information processing systems*, vol. 27, 2014.

[90] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[91] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[92] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[93] E. Bakopoulou, B. Tillman, and A. Markopoulou, "Fedpacket: A federated learning approach to mobile packet classification," *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3609–3628, 2022, doi: 10.1109/TMC.2021.3058627.

[94] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[95] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2021–2031.

[96] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[97] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[98] M. S. Al-Abiad, M. Z. Hassan, and M. J. Hossain, "Energy-efficient resource allocation for federated learning in noma-enabled and relay-assisted internet of things networks," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24 736–24 753, 2022.

[99] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE journal on selected areas in communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[100] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the gdpr perspective," *Computers & Security*, vol. 110, p. 102402, 2021.

[101] Q. Yang, L. Fan, and H. Yu, *Federated Learning: Privacy and Incentive*. Springer Nature, 2020, vol. 12500.

[102] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 1035–1044, 2021.

[103] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel, “Federated generalized face presentation attack detection,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022, doi:10.1109/TNNLS.2022.3172316.

[104] R. Shao, B. Zhang, P. C. Yuen, and V. M. Patel, “Federated test-time adaptive face presentation attack detection with dual-phase privacy preservation,” in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*. IEEE, 2021, pp. 1–8, doi:10.1109/FG52635.2021.9666952.

[105] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel, “Federated face presentation attack detection,” *arXiv preprint arXiv:2005.14638*, 2020, doi:10.1109/TNNLS.2022.3172316.

[106] L. Liu, Y. Zhang, H. Gao, X. Yu, and J. Cheng, “Fedfv: federated face verification via equivalent class embeddings,” *Multimedia Systems*, vol. 28, no. 5, pp. 1833–1843, 2022, doi:10.1007/s00530-022-00927-5.

[107] Y. Chen, L. Chen, C. Hong, and X. Wang, “Federated multitask learning with manifold regularization for face spoof attack detection,” *Mathematical Problems in Engineering*, vol. 2022, 2022, doi:10.1155/2022/7759410.

[108] M. Pei, B. Yan, H. Hao, and M. Zhao, “Person-specific face spoofing detection based on a siamese network,” *Pattern Recognition*, vol. 135, p. 109148, Mar 2023, doi:10.1016/j.patcog.2022.109148.

[109] X. Shu, X. Li, X. Zuo, D. Xu, and J. Shi, “Face spoofing detection based on multiscale color inversion dual-stream convolutional neural network,” *Expert Systems with Applications*, vol. 224, p. 119988, Aug 2023, doi:10.1016/j.eswa.2023.119988.

[110] C. Wang, B. Yu, and J. Zhou, “A learnable gradient operator for face presentation attack detection,” *Pattern Recognition*, vol. 135, p. 109146, Mar 2023, doi:10.1016/j.patcog.2022.109146.

[111] R. Huang and X. Wang, “Face anti-spoofing using feature distilling and global attention learning,” *Pattern Recognition*, vol. 135, p. 109147, Mar 2023, doi:10.1016/j.patcog.2022.109147.

[112] A. Günay Yılmaz, U. Turhal, and V. Nabiyev, “Face presentation attack detection performances of facial regions with multi-block LBP features,” *Multimedia Tools and Applications*, Mar 2023, doi:/10.1007/s11042-023-14453-7.

[113] H.-H. Chang and C.-H. Yeh, “Face anti-spoofing detection based on multi-scale image quality assessment,” *Image and Vision Computing*, vol. 121, p. 104428, May 2022, doi:10.1016/j.imavis.2022.104428.

[114] M. Fang, N. Damer, F. Kirchbuchner, and A. Kuijper, “Real masks and spoof faces: On the masked face presentation attack detection,” *Pattern Recognition*, vol. 123, p. 108398, Mar 2022, doi:10.1016/j.patcog.2021.108398.

[115] C. Wang, B. Yu, and J. Zhou, “A learnable gradient operator for face presentation attack detection,” *Pattern Recognition*, vol. 135, 2023, doi:10.1016/j.patcog.2022.109146.

[116] S. Fatemifar, S. Asadi, M. Awais, A. Akbari, and J. Kittler, “Face spoofing detection ensemble via multistage optimisation and pruning,” *Pattern Recognition Letters*, vol. 158, pp. 1–8, Jun 2022, doi:10.1016/j.patrec.2022.04.006.

[117] A. F. Ebihara, K. Sakurai, and H. Imaoka, “Efficient face spoofing detection with flash,” *IEEE Trans. Biom. Behav. Identity Sci.*, vol. 3, no. 4, pp. 535–549, Oct 2021, do:10.1109/TBIOM.2021.3076816.

[118] N. Daniel and A. Anitha, “Texture and quality analysis for face spoofing detection,” *Computers & Electrical Engineering*, vol. 94, p. 107293, Sep 2021, doi:10.1016/j.compeleceng.2021.107293.

[119] S. Jia, C. Hu, X. Li, and Z. Xu, “Face spoofing detection under super-realistic 3d wax face attacks,” *Pattern Recognition Letters*, vol. 145, pp. 103–109, May 2021, doi:10.1016/j.patrec.2021.01.021.

[120] W. Zhang and S. Xiang, “Face anti-spoofing detection based on DWT-LBP-DCT features,” *Signal Processing: Image Communication*, vol. 89, p. 115990, Nov 2020, doi:10.1016/j.image.2020.115990.

[121] A. George and S. Marcel, “Learning one class representations for face presentation attack detection using multi-channel convolutional neural networks,” *IEEE Transactions on Information Forensics and Security*, vol. 16, Jul 2020, doi:10.1109/TIFS.2020.3013214.

[122] W. Sun, Y. Song, C. Chen, J. Huang, and A. C. Kot, “Face spoofing detection based on local ternary label supervision in fully convolutional networks,” *IEEE Trans.Inform.Forensic Secur.*, vol. 15, pp. 3181–3196, 2020, doi:10.1109/TIFS.2020.2985530.

[123] X. Shu, H. Tang, and S. Huang, “Face spoofing detection based on chromatic ed-lbp texture feature,” *Multimedia Systems*, vol. 27, no. 2, pp. 161–176, Apr. 2021, doi:10.1007/s00530-020-00719-9.

[124] X. Song, X. Zhao, L. Fang, and T. Lin, “Discriminative representation combinations for accurate face spoofing detection,” *Pattern Recognition*, vol. 85, pp. 220–231, Jan 2019, doi :10.1016/j.patcog.2018.08.019.

[125] C. Yu, C. Yao, M. Pei, and Y. Jia, “Diffusion-based kernel matrix model for face liveness detection,” *Image and Vision Computing*, vol. 89, pp. 88–94, Sep 2019, doi : 10.1016/j.imavis.2019.06.009.

[126] A. George and S. Marcel, “Deep pixel-wise binary supervision for face presentation attack detection,” pp. 1–8, Jul 2019, doi:10.1109/ICB45273.2019.8987370.

[127] A. George, Z. Mostaani, D. Geissenbuhler, O. Nikisins, A. Anjos, and S. Marcel, "Biometric face presentation attack detection with multi-channel convolutional neural network," *IEEE Trans.Inform.Forensic Secur.*, vol. 15, pp. 42–55, 2020, doi:10.1109/TIFS. 2019.2916652.

[128] H. Chen, Y. Chen, X. Tian, and R. Jiang, "A cascade face spoofing detector based on face anti-spoofing r-cnn and improved retinex lbp," *IEEE Access*, vol. 7, pp. 170 116–170 133, 2019, doi:10.1109/ACCESS.2019.2955383.

[129] H. Chen, G. Hu, Z. Lei, Y. Chen, N. M. Robertson, and S. Z. Li, "Attention-based two-stream convolutional networks for face spoofing detection," *IEEE Trans.Inform.Forensic Secur.*, vol. 15, pp. 578–593, 2020, doi:10.1109/TIFS.2019.2922241.

[130] L. Li, P. L. Correia, and A. Hadid, "Face recognition under spoofing attacks: countermeasures and research directions," *IET biom.*, vol. 7, no. 1, pp. 3–14, Jan 2018, doi:10.1049/iet-bmt.2017.0089.

[131] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Trans.Inform.Forensic Secur.*, vol. 13, no. 7, pp. 1794–1809, Jul 2018, doi:10.1109/TIFS.2018.2801312.

[132] H. Li, P. He, S. Wang, A. Rocha, X. Jiang, and A. C. Kot, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Trans.Inform.Forensic Secur.*, vol. 13, no. 10, pp. 2639–2652, Oct 2018, doi:10.1109/TIFS.2018.2825949.

[133] L. Li, X. Feng, Z. Xia, X. Jiang, and A. Hadid, "Face spoofing detection with local binary pattern network," *Journal of Visual Communication and Image Representation*, vol. 54, pp. 182–192, Jul 2018, doi:10.1016/j.jvcir.2018.05.009.

[134] F. Xiong and W. AbdAlmageed, "Unknown presentation attack detection with face rgb images." IEEE, Oct 2018, pp. 1–9, doi:10.1109/BTAS.2018.8698574.

[135] I. Manjani, S. Tariyal, M. Vatsa, R. Singh, and A. Majumdar, "Detecting silicone mask-based presentation attack via deep dictionary learning," *IEEE Trans.Inform.Forensic Secur.*, vol. 12, no. 7, pp. 1713–1723, Jul 2017, doi:10.1109/TIFS.2017.2676720.

[136] P. P. K. Chan, W. Liu, D. Chen, D. S. Yeung, F. Zhang, X. Wang, and C.-C. Hsu, "Face liveness detection using a flash against 2d spoofing attack," *IEEE Trans.Inform.Forensic Secur.*, vol. 13, no. 2, pp. 521–534, Feb 2018, doi:10.1109/TIFS.2017.2758748.

[137] S. R. Arashloo, J. Kittler, and W. Christmas, "An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol," *IEEE Access*, vol. 5, pp. 13 868–13 882, 2017, doi:url 10.1109/ACCESS.2017.2729161.

[138] G. B. de Souza, D. F. da Silva Santos, R. G. Pires, A. N. Marana, and J. P. Papa, "Deep texture features for robust face spoofing detection," *IEEE Trans. Circuits Syst. II*, vol. 64, no. 12, pp. 1397–1401, Dec. 2017, doi:10.1109/TCSII.2017.2764460.

[139] F. Peng, L. Qin, and M. Long, "Face presentation attack detection using guided scale texture," *Multimed Tools Appl*, vol. 77, no. 7, pp. 8883–8909, Apr 2018, doi:10.1007/s11042-017-4780-0.

[140] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Trans.Inform.Forensic Secur.*, vol. 11, no. 8, pp. 1818–1830, Aug 2016, doi:10.1109/TIFS.2016.2555286.

[141] Q.-T. Phan, D.-T. Dang-Nguyen, G. Boato, and F. G. B. De Natale, "Face spoofing detection using ldp-top." IEEE, Sep 2016, pp. 404–408, doi:10.1109/ICIP.2016.7532388.

[142] H. Li, S. Wang, and A. C. Kot, "Face spoofing detection with image quality regression." IEEE, Dec 2016, pp. 1–6, doi:10.1109/IPTA.2016.7821027.

[143] A. Ali, S. Hoque, and F. Deravi, "Gaze stability for liveness detection," *Pattern Anal Applic*, vol. 21, no. 2, pp. 437–449, May 2018, doi:10.1007/s10044-016-0587-2.

[144] T. A. Siddiqui, S. Bharadwaj, T. I. Dhamecha, A. Agarwal, M. Vatsa, R. Singh, and N. Ratha, "Face anti-spoofing with multifeature videolet aggregation," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, Dec 2016, pp. 1035–1040, doi:10.1109/ICPR.2016.7899772.

[145] A. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha, "Face spoofing detection through visual codebooks of spectral temporal cubes," *IEEE Trans. on Image Process.*, vol. 24, no. 12, pp. 4726–4740, Dec 2015, doi:10.1109/TIP.2015.2466088.

[146] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Trans.Inform.Forensic Secur.*, vol. 10, no. 4, pp. 864–879, Apr 2015, doi:10.1109/TIFS.2015.2398817.

[147] Di Wen, Hu Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Trans.Inform.Forensic Secur.*, vol. 10, no. 4, pp. 746–761, Apr 2015, doi:10.1109/TIFS.2015.2400395.

[148] Z. Boulkenafet, J. Komulainen, and A. Hadid, "face anti-spoofing based on color texture analysis," *2015 IEEE International Conference on Image Processing (ICIP)*, Nov 2015, doi:10.1109/ICIP.2015.7351280.

[149] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. S. Ho, "Detection of face spoofing using visual dynamics," *IEEE Trans.Inform.Forensic Secur.*, vol. 10, no. 4, pp. 762–777, Apr 2015, doi:10.1109/TIFS.2015.2406533.

[150] A. Anjos, M. M. Chakka, and S. Marcel, "Motion-based counter-measures to photo attacks in face recognition," *IET biometrics*, vol. 3, no. 3, pp. 147–158, 2014, doi:10.1049/iet-bmt.2012.0071.

[151] R. Raghavendra and C. Busch, "Novel presentation attack detection algorithm for face recognition system: Application to 3d face mask attack," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, Oct 2014, pp. 323–327, doi:10.1109/ICIP.2014.7025064.

[152] J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," in *2014 22nd International Conference on Pattern Recognition*. IEEE, Aug 2014, pp. 1173–1178, doi:10.1109/ICPR.2014.211.

[153] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition," *IEEE Trans. on Image Process.*, vol. 23, no. 2, pp. 710–724, Feb 2014, doi:10.1109/TIP.2013.2292332.

[154] N. Erdogmus and S. Marcel, "Spoofing face recognition with 3d masks," *IEEE Trans.Inform.Forensic Secur.*, vol. 9, no. 7, pp. 1084–1097, Jul 2014, doi:10.1109/TIFS.2014.2322255.

[155] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, Jun 2013, pp. 105–110, doi:10.1109/CVPRW.2013.23.

[156] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?" in *2013 International Conference on Biometrics (ICB)*. IEEE, Jun 2013, pp. 1–8, doi:10.1109/ICB.2013.6612981.

[157] N. Erdogmus and S. Marcel, "Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, Sep 2013, pp. 1–6, doi:10.1109/BTAS.2013.6712688.

[158] C.-L. Lai, J.-H. Chen, J.-Y. Hsu, and C.-H. Chu, "Spoofing face detection based on spatial and temporal features analysis," in *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)*. IEEE, Oct 2013, pp. 301–302, doi:10.1109/GCCE.2013.6664836.

[159] T. Wang, J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection using 3d structure recovered from a single camera," in *2013 International Conference on Biometrics (ICB)*. IEEE, Jun 2013, pp. 1–6, doi:10.1109/ICB.2013.6612957.

[160] J. Maatta, A. Hadid, and M. Pietikainen, "Face spoofing detection from single images using micro-texture analysis," in *2011 International Joint Conference on Biometrics (IJCB)*. IEEE, Oct 2011, pp. 1–7, doi:10.1109/IJCB.2011.6117510.

[161] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: A public database and a baseline," in *2011 International Joint Conference on Biometrics (IJCB)*. IEEE, Oct 2011, pp. 1–7, doi:10.1109/IJCB.2011.6117503.

[162] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using texture and local shape analysis," *IET Biometrics*, vol. 1, no. 1, p. 3 – 10, Mar 2012, doi:10.1049/iet-bmt.2011.0009.

[163] M. De Marsico, M. Nappi, D. Riccio, and J.-L. Dugelay, "Moving face spoofing detection via 3d projective invariants," in *2012 5th IAPR International Conference on Biometrics (ICB)*. IEEE, Mar 2012, pp. 73–78, doi:10.1109/ICB.2012.6199761.

[164] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," *2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*, pp. 1–7, Sep 2012.

[165] H. M. Khan and P. Venkadesh, "Spoofing free fingerprint image enhancement," *International Journal of Information Technology*, vol. 15, no. 1, pp. 477–485, 2023, doi:10.1007/s41870-022-01129-y.

[166] S. M. Abdullahi, S. Sun, A. Malik, O. Khudeyberdiev, and R. Basheer, "Spoofed fingerprint image detection using local phase patch segment extraction and a lightweight network," in *IFIP International Conference on Digital Forensics*. Springer, 2022, pp. 85–105, doi:10.1007/978-3-031-10078-9_5.

[167] M. Saguy, J. Almog, D. Cohn, and C. Champod, "Proactive forensic science in biometrics: Novel materials for fingerprint spoofing," *Journal of Forensic Sciences*, vol. 67, no. 2, pp. 534–542, 2022, doi:10.1111/1556-4029.14908.

[168] N. Pałka and M. Kowalski, "Towards fingerprint spoofing detection in the terahertz range," *Sensors*, vol. 20, no. 12, p. 3379, 2020, doi:10.3390/s20123379.

[169] S. Arora and M. S. Bhatia, "Fingerprint spoofing detection to improve customer security in mobile financial applications using deep learning," *Arabian journal for science and engineering*, vol. 45, no. 4, pp. 2847–2863, 2020, doi:10.1007/s13369-019-04190-1.

[170] G. B. de Souza, D. F. da Silva Santos, R. G. Pires, A. N. Marana, and J. P. Papa, "Deep features extraction for robust fingerprint spoofing attack detection," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 1, pp. 41–49, 2019, doi:10.2478/jaiscr-2018-0023.

[171] A. Toosi, S. Cumani, and A. Bottino, "Assessing transfer learning on convolutional neural networks for patch-based fingerprint liveness detection," in *Computational Intelligence: 9th International Joint Conference, IJCCI 2017 Funchal-Madeira, Portugal, November 1-3, 2017 Revised Selected Papers*. Springer, 2019, pp. 263–279, doi:10.1007/978-3-030-16469-0_14.

[172] 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*, vol. 7, pp. 91 476–91 487, 2019, doi:10.1109/ACCESS.2019.2927357.

[173] G. B. Souza, D. F. Santos, R. G. Pires, A. N. Marana, and J. P. Papa, "Deep boltzmann machines for robust fingerprint spoofing attack detection," in *2017 International Joint Conference on Neural Networks (IJCNN).* Ieee, 2017, pp. 1863–1870, doi:10.1109/IJCNN.2017.7966077.

[174] C. Yuan, X. Li, Q. J. Wu, J. Li, and X. Sun, "Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis," *Computers, Materials & Continua*, vol. 53, no. 3, pp. 357–371, 2017, doi:10.3970/cmc.2017.053.357.

[175] A. Balaji, V. HS, and S. OK, "Multimodal fingerprint spoof detection using white light," *Procedia Computer Science*, vol. 78, pp. 330–335, 2016, doi:10.1016/j.procs.2016.02.066.

[176] P. Wild, P. Radu, L. Chen, and J. Ferryman, "Robust multimodal face and fingerprint fusion in the presence of spoofing attacks," *Pattern Recognition*, vol. 50, pp. 17–25, 2016, doi:10.1016/j.patcog.2015.08.007.

[177] E. Park, W. Kim, Q. Li, J. Kim, and H. Kim, "Fingerprint liveness detection using cnn features of random sample patches," in *2016 International Conference of the Biometrics Special Interest Group (BIOSIG).* IEEE, 2016, pp. 1–4, doi:10.1109/BIOSIG.2016.7736923.

[178] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 864–879, 2015, doi:10.1109/TIFS.2015.2398817.

[179] A. Rattani, W. J. Scheirer, and A. Ross, "Open set fingerprint spoof detection across novel fabrication materials," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2447–2460, 2015, doi:10.1109/TIFS.2015.2464772.

[180] Z. Akhtar, C. Michelon, and G. L. Foresti, "Liveness detection for biometric authentication in mobile applications," in *2014 International Carnahan Conference on Security Technology (ICCST).* IEEE, 2014, pp. 1–6, doi:10.1109/CCST.2014.6986982.

[181] Z. Akhtar, C. Micheloni, C. Piciarelli, and G. L. Foresti, "Mobio_livdet: Mobile biometric liveness detection," in *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* IEEE, 2014, pp. 187–192, doi:10.1109/AVSS.2014.6918666.

[182] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI 11*, 2010, pp. 504–517.

[183] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[184] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic,” *Measurement*, vol. 167, p. 108288, 2021.

[185] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[186] M. Toğaçar, B. Ergen, and Z. Cömert, “Covid-19 detection using deep learning models to exploit social mimic optimization and structured chest x-ray images using fuzzy color and stacking approaches,” *Computers in biology and medicine*, vol. 121, p. 103805, 2020.

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

[188] E. Deniz, A. Şengür, Z. Kadiroğlu, Y. Guo, V. Bajaj, and Ü. Budak, “Transfer learning based histopathologic image classification for breast cancer detection,” *Health information science and systems*, vol. 6, pp. 1–7, 2018.

[189] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, “Communication-efficient learning of deep networks from decentralized data.”

[190] Q. u. Ain, M. A. Khan, M. M. Yaqoob, U. F. Khattak, Z. Sajid, M. I. Khan, and A. Al-Rasheed, “Privacy-aware collaborative learning for skin cancer prediction,” *Diagnostics*, vol. 13, 2023, doi: 10.3390/diagnostics13132264.

[191] Y. Chen, L. Chen, C. Hong, and X. Wang, “Federated multitask learning with manifold regularization for face spoof attack detection ,” vol. 2022, pp. 1–10, doi: 10.1155/2022/7759410.

[192] X. Tan, Y. Li, J. Liu, and L. Jiang, “Face liveness detection from a single image with sparse low rank bilinear discriminative model,” in *Computer Vision – ECCV 2010*. Springer Berlin Heidelberg, pp. 504–517.

[193] J. Komulainen, A. Hadid, and M. Pietikäinen, “Face spoofing detection using dynamic texture,” ser. Computer Vision - ACCV 2012 Workshops. Springer Berlin Heidelberg, pp. 146–157.

[194] J. Li, Y. Wang, T. Tan, and A. K. Jain, “Live face detection based on the analysis of fourier spectra,” in *Biometric Technology for Human Identification*, vol. 5404. SPIE, 2004, pp. 296 – 303.

[195] S. Parveen, S. M. S. Ahmad, N. H. Abbas, W. A. W. Adnan, M. Hanafi, and N. Naeem, “Face liveness detection using dynamic local ternary pattern (dltp),” *Computers*, vol. 5, 2016.

[196] X. Liu, R. Lu, and W. Liu, “Face liveness detection based on enhanced local binary patterns,” in *2017 Chinese Automation Congress (CAC)*. IEEE, pp. 6301–6305.

[197] R. Koshy and A. Mahmood, “Optimizing deep CNN architectures for face liveness detection,” vol. 21, no. 4, p. 423.

[198] V. Mura, L. Ghiani, G. L. Marcialis, F. Roli, D. A. Yambay, and S. A. Schuckers, “Livdet 2015 fingerprint liveness detection competition 2015,” 2015.

BIOGRAPHICAL SKETCH

S M Sarwar received his Bachelor of Science and Master of Science degrees in Computer Science and Engineering from Jahangirnagar University in Bangladesh, respectively, in Octo-ber 2019 and March 2022. Later, he received his Master of Science in Computer Science from the University of Texas Rio Grande Valley (UTRGV) in August 2023.

After joining UTRGV in Fall 2021, he started working on Privacy-Preserving Machine Learning (Federated learning) as a Graduate Research Assistant (GRA) and an awardee of the Presidential Research Fellowship (PRF) in August 2021. Outside of the classroom, he was elected Secretary of Engineers Without Borders (EWB), UTRGV Chapter. In Spring 2022, he was elected Vice President of the Collegiate Entrepreneurs' Organization (CEO), UTRGV Chapter, and in Summer 2022, he became its President. In a short time, S M made a tremendous impact in leading the organization and preparing the CEO - UTRGV chapter for the future. In addition, S M was elected as a Graduate Senator for UTRGV's Student Government Association (SGA) for the 2022–2023 academic year. He was a member of the academic affairs committee of SGA and he closely worked to establish a Graduate Students Council (GSC). However, he was an alternate representative of the UT System Student Advisory Council (UT SAC) for 2022–23.

As a student at UTRGV, he participated in NeurIPS'22, ICML'22, FCCM'22, PPML'22, SOUPS-USENIX'22, HSI Battle of the Brain'22, UTRGV NSF I-Corps Program'22, CEO Global Conference'22, UTRGV Engaged Scholar Symposium'22, and some other research-related conferences.

Outside of school, he volunteered to organize events and track metrics as a Mozilla Representative for two Mozilla communities in India. Furthermore, starting in the Summer 2022, he started working as a Vice-Chair at the IEEE Corpus Christi Section as a community service and continues to work as the Vice-Chair today. He was a graduate student member of the IEEE, IEEE Computer Society, AAAI, ACM and SHPE.

His email address is smsarwar96@gmail.com. Anyone seeking support or details may reach him at the following address.