ML-CHIEFS: MACHINE LEARNING-BASED CORNEAL-SPECULAR HIGHLIGHT IMAGING FOR ENHANCING FACIAL RECOGNITION SECURITY

A Dissertation
IN
Computer Science
and
Computer Networking and Communication Systems

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

DOCTOR OF PHILOSOPHY

by
MUHAMMAD ALI H MOHZARY

M. S., Kent State University, Kent, Ohio, USA, 2018
B. S., Jazan University, Jazan, Saudi Arabia, 2013

Kansas City, Missouri
2023

ML-CHIEFS: MACHINE LEARNING-BASED CORNEAL-SPECULAR HIGHLIGHT IMAGING FOR ENHANCING FACIAL RECOGNITION SECURITY

Muhammad Ali H Mohzary, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2023

ABSTRACT

Machine learning (ML) has significantly improved facial recognition systems' (FRS) accuracy, robustness, and reliability, making them one of the most viable biometric identity verification solutions in various authentication applications. However, there are concerns about using FRSs, including privacy violations, fake presentations, potential biases, and security issues. Furthermore, the remarkable advancement of AI-leveraged production and manipulation techniques of fictitious human facial images, DeepFake, elevates spreading misinformation and creating deception for identity theft, which becomes critical security and privacy threat. In this dissertation, ***ML-CHIEFS***: ***M***achine ***L***earning-based ***C***orneal-specular ***H***ighlight ***I***maging for ***E***nhancing ***F***acial recognition ***S***ecurity, we researched and developed unique technologies to resolve significant FRS challenges, including Deepfakes and identity thefts, liveness presentation attacks (PA), and master face dictionary attacks (MFDA). We propose countermeasures against facial

biometric PAs, detect DeepFakes, and identify MFDA using intelligent ML-based specular highlights detections upon the hypothesis that the existing facial spoofings fail to coordinate their counterfeits with the reflective components. First, we designed a software-based facial liveness detection method named Apple in My Eyes (AIME). AIME is intended to detect the liveness against spoofing for mobile device security using challenge-response testing. Our comprehensive experimental results reveal that AIME can efficiently detect PAs with high accuracy at around 200 ms against different types of sophisticated presentation attacks without any costly extra sensors nor involving users' active responses. Second, we proposed novel ML-based DeepFake detection technologies, including CHIEFS (Corneal-Specular Highlights Imaging for Enhancing Fake-Face Spotter), MobiDeep (Mobile DeepFake Detection through ML-based Corneal-Specular Backscattering), and READFake (Reflection and Environment-Aware DeepFake). CHIEFS detects various corneal-specular and facial highlights features and inspects the ensemble of the highlights with the surrounding environmental factors. The empirical results show that it improves the detection accuracy from 86.05% with the reflection shape similarity alone to 99.00% with the ResNet-50-V2 architecture. MobiDeep is a real-time, cloudless,

lightweight mobile application for human visual DeepFake detection using ML technologies, which achieved high accuracy (98.7%) and rapid detection speed in detecting sophisticated DeepFake images within 200 ms. The READFake detection technique uses specular highlights on various facial and body parts and environmental factors. We have conducted extensive experiments to evaluate the performance of READFake using different input parameters and advanced DNN architectures on multiple public DeepFake datasets. The experimental results indicate that READFake achieves better accuracy (99.0%) than the SOTA methods in detecting DeepFake images. Finally, we develop a novel countermeasure against MFDAs using a Reflection-based Identification (DARI) system. Using a lightweight and low-latency vision transformer, we build a feature extractor network to identify the inconsistencies among the facial image's specular highlights and physiological characteristics. The empirical results show that DARI achieves very high detection accuracy ranging from 97.83% to 99.56% on public GAN-face detection datasets and instantaneous detection speed (less than 11 ms) against SOTA master face dictionary attacks.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled "ML-CHIEFS: Machine Learning-based Corneal-specular Highlight Imaging for Enhancing Facial Recognition Security," presented by Muhammad Ali H Mohzary, candidate for the Doctor of Philosophy degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Sejun Song, Ph.D., Committee Chair and Primary Discipline Advisor
Department of Computer Science & Electrical Engineering

Baek-Young Choi, Ph.D., Co-discipline Advisor
Department of Computer Science & Electrical Engineering

Cory Beard, Ph.D.
Department of Computer Science & Electrical Engineering

Zhu Li, Ph.D.
Department of Computer Science & Electrical Engineering

Praveen Rao, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and appreciation to my academic advisors, Dr. Sejun Song and Dr. Beak-Young Choi, for their support, guidance, and expertise throughout my doctoral journey. Their invaluable feedback, constructive criticism, and encouragement have been instrumental in shaping the research and analysis presented in this dissertation. I am grateful for their patience, dedication, and mentorship, and I feel fortunate to have had the opportunity to work with them.

I want to express my sincere appreciation to the members of my dissertation committee, Dr. Cory Beard, Dr. Zhu Li, and Dr. Praveen Rao. I want to thank them all for their time, effort, and commitment. I am honored to have had the privilege to work with such a knowledgeable and dedicated committee.

I am immensely grateful for the support and encouragement I received from my friends and colleagues at the School of Science and Engineering and the Trustworthy Systems and Software Research lab at UMKC. I want to extend my heartfelt thanks to Dr. Khalid J Almalki, Dr. Abdoh Jabbari, Dr. Kaushik Ayinala, Md Tajul Islam, Samuel Akinyede, Dr. Mohamed Gharibi, Dr. Nouf Alrasheed, Syed Jawad H Shah, and Dr. Ahmed Albishri, for their support throughout my academic journey. I am fortunate to have had such amazing colleagues who were always willing to lend a helping hand. Thank you all for your support and friendship.

Thanks should also go to my wife, Aeshah Mahzari, my son, Ali, and my daughter, Lama. Their constant love, patience, and understanding have been the driving force

behind my success, and I am forever grateful for their presence in my life.

I want to thank my parents, Ali and Fatima Mohzary, my grandparents, and my brothers and sisters for their endless support. I want to acknowledge the sacrifices my family has made during this journey, and I am grateful for their understanding and support. Without their love and support, I would not have been able to complete this Ph.D. program. Additionally, I would like to sincerely thank my uncle Dr. Ibrahim Mohzary, my cousin Dr. Qassim Mohzary, and my brother-in-law Dr. Abdulkarim Malkadi for their support.

I am indebted to Jazan University and my home country, Saudi Arabia, for their generous financial support throughout my Ph.D. program. I am deeply grateful for their investment in my education and the opportunities their sponsorship has afforded me. Without their scholarship, I would not have been able to reach this stage of my academic journey. I appreciate their contributions to my accomplishments, and I hope to make a positive impact in my country and beyond with the knowledge and skills I have gained.

PART 1

OVERVIEW

CHAPTER 1

INTRODUCTION

1.1 An Overview of Biometric Authentications

Passwords have been a staple of online security for decades. However, in recent years they have increasingly become less effective for authentication due to various factors. One major issue is that passwords are the primary cause of more than 80% of data breaches because people tend to choose weak and easily guessable passwords, which hackers can easily crack [27]. This is especially problematic given the prevalence of large-scale data breaches in recent years, which have exposed billions of user-sensitive information.

According to Fast Identity Online (FIDO) Alliance [27], up to 51% of passwords are reused. Thus, even if users choose strong passwords, they often reuse them across multiple accounts, which can lead to a domino effect where one compromised password can compromise multiple accounts.

Another issue with passwords is that they are vulnerable to phishing attacks, where attackers trick users into revealing their passwords by impersonating legitimate websites or services. Phishing attacks can be challenging to detect, especially when they are well-crafted, and can lead to the theft of sensitive information and account takeover [35].

In addition to these human factors, passwords are also vulnerable to technical weaknesses. For example, passwords can be intercepted by malware running on a user's

computer or transmitted in plaintext over unsecured networks. They can also be brute-forced by attackers with access to powerful computing resources or by exploiting weaknesses in the authentication system [35].

To address these weaknesses, many organizations have begun implementing additional forms of authentication, such as multi-factor or biometric authentications. Multi-factor authentications require users to provide two or more types of authentication credentials, such as a password and a one-time code sent to their mobile device, which can provide a much higher level of security than passwords alone [33].

Biometric authentication, which uses physiological characteristics like fingerprints, facial recognition, or iris scans, can also provide a high level of security, as these characteristics are unique to each individual and difficult to replicate [65]. Biometrics are becoming increasingly popular as a replacement for password-only logins because of their secure and fast login experiences across websites and apps, capabilities to mitigate data breach risks and damages, and enhanced user experience [65].

The COVID-19 pandemic also has led to a surge in demand for contactless biometric authentication methods as people have become more concerned about the potential spread of the virus through physical contact. Several popular touch-based authentication methods, such as fingerprint scanning or hand geometry, were heavily restricted for perceptions of safety as impacts of Coronavirus. As a result, contactless biometric authentication methods have gained popularity as a safer alternative. In addition to reducing the risk of virus transmission, contactless biometric authentication methods also offer other benefits. They are often faster and more convenient than traditional methods, as users do

not need to physically interact with a device or surface, reducing the likelihood of errors or delays. Contactless methods are also more hygienic and require less maintenance compared to traditional methods [84]. The COVID-19 pandemic has accelerated the adoption of contactless biometric authentication methods in many organizations. As a result, the global contactless biometrics market was valued at USD 11.8 billion in 2020, grew to USD 13.45 billion in 2022, and is expected to reach USD 78.86 billion by 2032, growing at a compound annual growth rate of 19.35% [4].

Facial recognition system (FRS) is currently the most popular contactless biometric authentication [182]. FRS is not only contactless but works with all devices people use daily. It is also fast, convenient, and works across all services. FRS is now widely available in many consumer electronic devices, including smartphones, laptops, and tablets. This has made it a popular authentication method for mobile device users who want a convenient and secure way to access their devices.

FRS is a technology that can identify individuals based on their facial features, usually using images or video footage captured by cameras. In authentication and identity verification, FRS confirms persons' identities by comparing their facial features to a previously stored template.

FRS works in authentication and identity verification by following three basic steps:

1. Enrollment, the FRS captures an image of the user's face and creates a digital template that is stored in a database.

2. Authentication, so when the user tries to access a system or service, the FRS captures another image of the face and compares it to the stored template.

3. Verification, if the facial features in the new image match those in the stored template within a certain degree of accuracy, the user is verified and granted access to the system or service.

FRS used in authentication and identity verification offers several benefits. For instance, FRS can enhance security by making it difficult for unauthorized users to access sensitive information or services. In addition, it can provide a convenient and seamless user experience by eliminating the need for passwords or other authentication methods. FRS also can help prevent identity theft and other types of fraud by verifying the user's identity in real-time. Furthermore, FRS can process authentication and identity verification requests quickly and accurately, which can save time and resources.

Overall, FRSs have the potential to improve security and enhance user experience in authentication and identity verification. However, their use should be carefully evaluated to ensure that they do not infringe on individuals' security or privacy.

1.2 Face Spoofing Attacks

FRSs offer several benefits, including enhanced security, personalized experiences, and convenient authentication. Nevertheless, significant security and privacy challenges are also associated with these systems. Some of the critical challenges are FRSs' vulnerability to face spoofing attacks, which involve presenting the system with a fake

or manipulated image of a face to impersonate someone else and gain unauthorized access to a system or data. Face spoofing attacks can be performed using various methods, including print attacks, replay attacks, 3D mask attacks, DeepFakes attacks, and master face dictionary attacks (MFDA).

1.2.1 Presentation Attacks (PA)

Presentation attacks (PA) are a type of biometric attack where an unauthorized person attempts to gain access to a system or information by impersonating a legitimate user through the use of fake or altered facial features. This is usually done by presenting a counterfeit image or video of the user's face, such as a printed photograph, a digital image, or a video recording, to FRS. The goal of the attacker is to bypass the security measures that rely on facial recognition technology, which may include authentication systems for accessing a computer or mobile device, entering secure areas, or making financial transactions [34]. As shown in Figures 1 and 2, there are different types of face PAs, including:

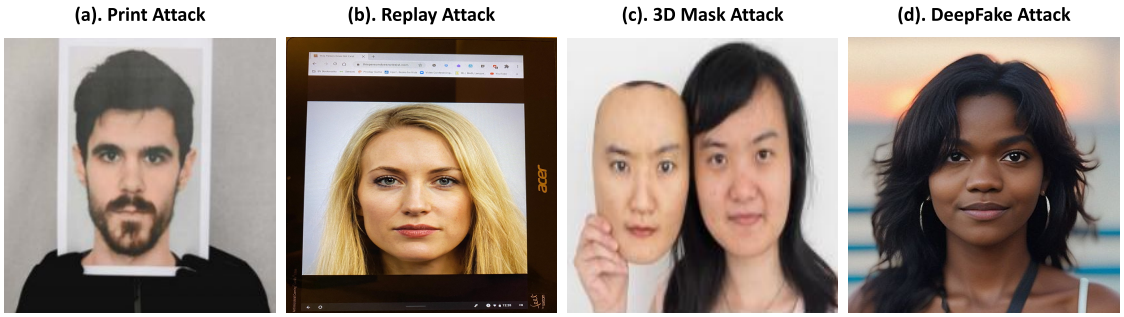


Figure 1: Face presentation attacks: (a) print attack [76], (b) replay attack, (c) 3D mask attack [107], and (d) DeepFake attack [17].

1. Print Attack: In this attack, an attacker uses a printed image of a user's face to impersonate them.

2. Replay Attack: In this type of attack, an attacker uses a recorded video of a user's face to impersonate them.

3. 3D Mask Attack: In this type of attack, an attacker uses a 3D printed or molded mask of a user's face to impersonate them.

4. DeepFake Attack: In this type of attack, an attacker uses artificial intelligence (AI) and machine learning (ML) algorithms to create a realistic fake video of a user's face to impersonate them.

5. Camouflage Attacks: Face camouflage attacks are a type of biometric attack where an attacker tries to evade FRSs by disguising their facial features using makeup, accessories, or other techniques to alter their appearance. This attack aims to fool the FRSs into thinking that the attacker is a different person or to make it difficult for the systems to identify the attacker accurately [139, 174]. Figure 2 shows some common techniques used in face camouflage attacks, including:

   (a) Makeup: The attacker can use makeup to change the appearance of their face, such as applying dark shadows to change the shape of their eyes or applying foundation to alter the skin tone.

   (b) Wigs or Hats: The attacker can wear a wig or hat to change the shape of their hairline or the size of their forehead.

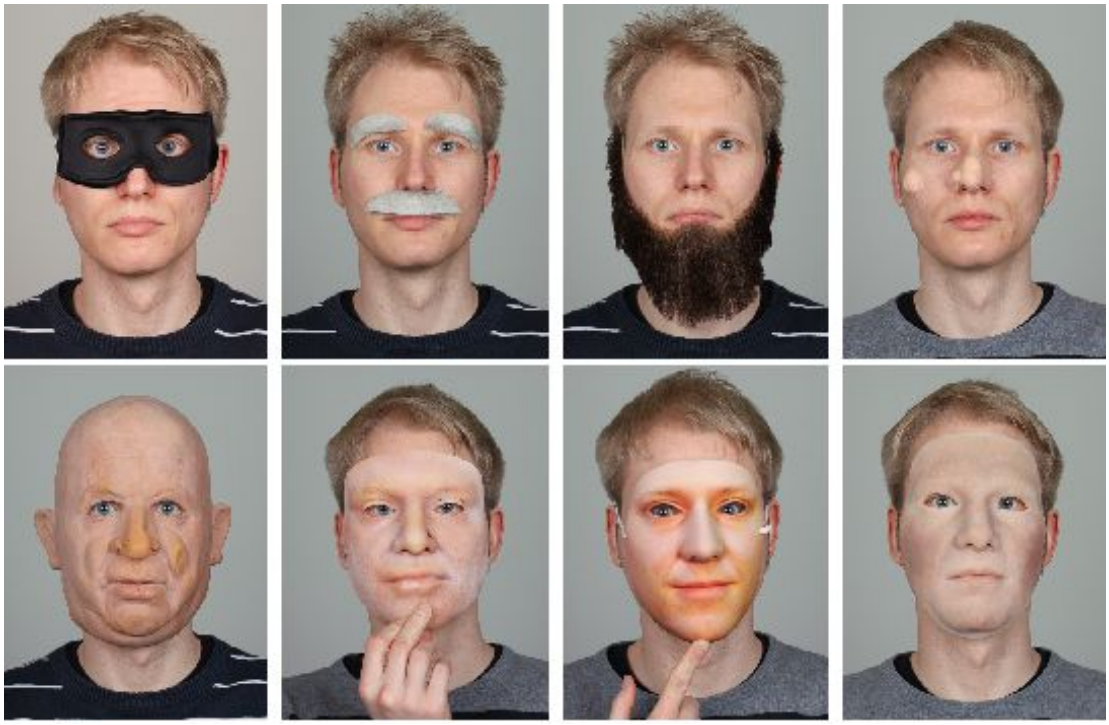


Figure 2: Face camouflage attacks [158].

(c) Glasses: The attacker can wear glasses to obscure their eyes and eyebrows, making it difficult for the facial recognition system to detect their facial features.

(d) Masks: The attacker can wear a mask or other facial covering to hide their face from the FRSs completely.

6. Adversarial Attacks: Adversarial attacks are a type of attack that is used against FRSs to avoid or trick them by introducing small perturbations or changes to the input data. Adversarial attacks can be used to bypass or misclassify the input data, leading to incorrect recognition results. The main idea behind adversarial attacks

is to generate small perturbations that are not noticeable to the human eye but can significantly change the input data in a way that the FRS misclassifies the input. These perturbations can be added to the image, or the attacker can change the lighting conditions or the camera angle to generate the perturbations. There are different types of adversarial attacks used against FRSs [168], such as:

(a) Targeted Attacks: In this type of attack, the attacker has a specific target in mind, and they generate the adversarial perturbations to misclassify the input data as that specific target.

(b) Non-targeted Attacks: In this type of attack, the attacker does not have a specific target in mind, and they generate the adversarial perturbations to make the FRS misclassify the input data as something other than the correct identity.

(c) Transfer Attacks: In this type of attack, the attacker generates the adversarial perturbations on one FRS and then transfers them to another system to test the system's robustness and generalizability.

To prevent face PAs, security measures, such as liveness detection, anti-spoofing techniques, and combining different biometric authentication factors can increase the security of FRSs and reduce the risk of face presentation spoofing attacks [34].

1.2.2 DeepFakes Attacks

DeepFake is a type of synthetic media that uses AI and deep learning techniques to create realistic fake videos or images. The term "DeepFake" is a combination of "deep learning" and "fake." DeepFakes can be created by training ML models on large datasets

of images and videos of a person's face, voice, or body movements. These models can then generate new images or videos that appear to be of the same person but are actually entirely fabricated [91].

The implications of DeepFakes are significant and far-reaching. One of the biggest concerns is their potential to be used for malicious purposes, such as spreading misinformation, committing fraud, or creating fake pornography. For example, a DeepFake video of a politician saying or doing something inappropriate or illegal could be used to damage their reputation or sway public opinion. DeepFakes can also be used to impersonate someone, such as creating a fake video or audio message from a CEO to trick employees into revealing sensitive information. Another concern is the impact of DeepFakes on trust and credibility. As DeepFakes become more sophisticated and harder to detect, it becomes more challenging to determine what is real and what is fake. This can erode public trust in media, undermine the credibility of information sources, and fuel conspiracy theories and misinformation [31, 63, 91].

To address these concerns, researchers are developing new methods to detect DeepFakes and identify their sources. However, detecting DeepFakes can be difficult and time-consuming. Furthermore, there is no guarantee that detection methods will keep up with the rapid advancements in DeepFake technology. Overall, deepfakes represent a significant challenge for society and require careful consideration and action to mitigate their negative impacts [31].

1.2.3 Master Face Spoofing Attacks (MFDA)

MFDA is a type of cyber attack that targets biometric FRSs. In this type of attack, the attacker creates a database of facial images, which is known as the ”master face dictionary.” The attacker then uses this database to gain unauthorized access to FRS by presenting a fake or modified image of someone else's face [129, 154]. The process of creating a master face dictionary typically involves the following steps:

- Collecting facial images: The first step is to collect a large number of facial images. These images can be obtained from public sources such as social media or through more nefarious means, such as hacking into a database of images.
- Pre-processing images: Once the images have been collected, they are pre-processed to enhance their quality and remove any noise or distortions that could affect creating a master face dictionary.
- Creating templates: The next step is to create templates from the facial images. Templates are essentially mathematical representations of a person's face that can be used to compare against other facial images.
- Storing templates: The templates are then stored in a database or ”master face dictionary” that can be used in a subsequent attack on FRSs.
- Modifying images: In some cases, attackers may modify the facial images to improve their chances of bypassing the FRSs. For example, they may alter the lighting or angle of the image or add accessories such as glasses or hats to the face.

Once the master face dictionary has been created, the attacker can try to gain unauthorized access to FRS by presenting a fake or modified image from the master face dictionary. Suppose the system is not able to detect that the image is not a genuine representation of the person's face. In that case, the attacker can successfully bypass the system.

The implications of MFDA can be significant. If the FRSs are compromised, it can result in unauthorized access to sensitive information or locations. For example, if the attack successfully compromises FRSs to access a secure facility or authenticate financial transactions, it could lead to theft or other criminal activity. Moreover, FRSs are increasingly being used in law enforcement, border control, and other security-related applications. If these systems are compromised, it could have serious implications for public safety.

To mitigate the risk of MFDA, it is important to ensure that FRSs incorporate robust security measures such as strong authentication, multi-factor authentication implementation, and liveness detection technologies [129].

1.3 An Overview of Light Specularity

Light specularity refers to the phenomenon of light reflecting off a surface in a mirror-like manner. Objects can exhibit a range of colors due to the varying wavelengths of light that their surfaces reflect. Furthermore, the shininess of objects can vary based on the directions in which their surfaces reflect light [165].

When light strikes a surface, it can be absorbed, transmitted, or reflected. When

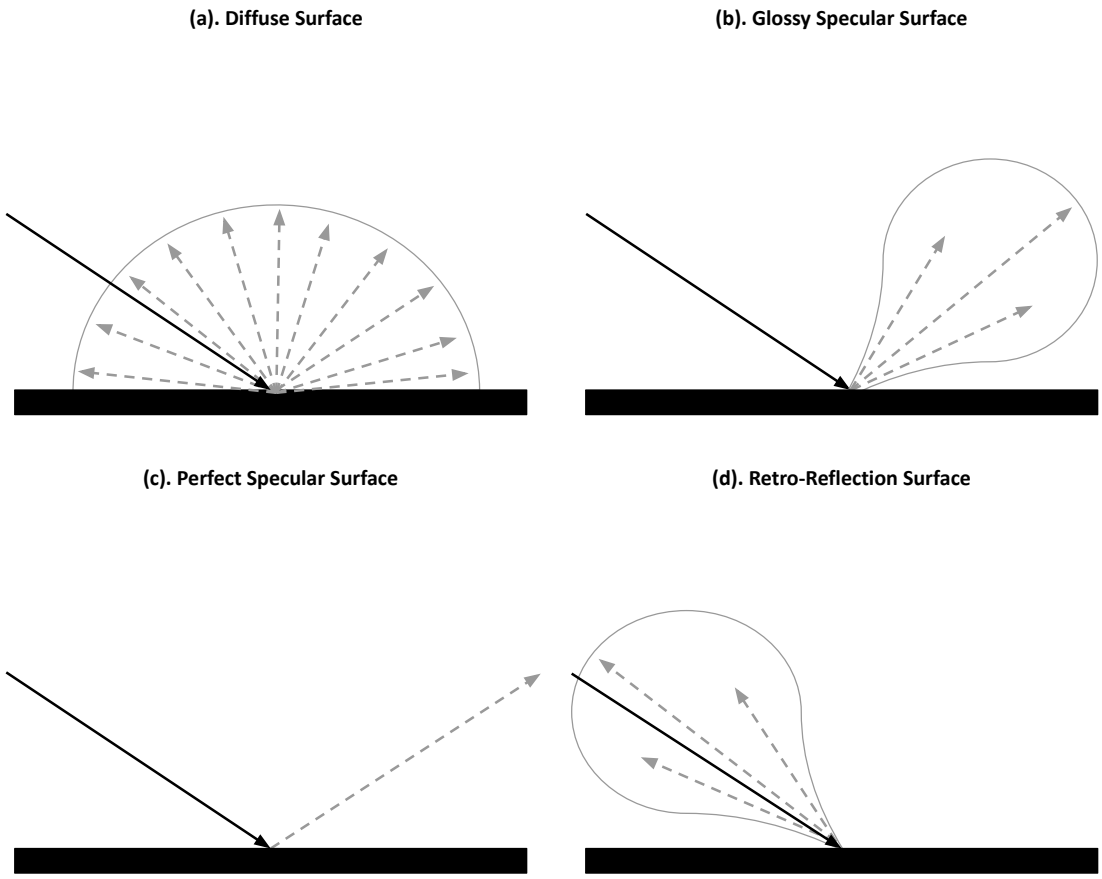


Figure 3: Types of surfaces and reflected light directions.

it is reflected, the angle of incidence (the angle between the incident light and the surface normal) and the angle of reflection (the angle between the reflected light and the surface normal) are equal. As demonstrated in Figure 3, four types of surfaces reflect light over a different range of directions. In each case, the light ray is shown in black and the reflected ray(s) are shown in gray. For example, the diffuse surfaces in Figure 3 (a) reflect light equally in all directions. The glossy specular surfaces in Figure 3 (b) reflect light over a limited range of directions. The perfect specular surfaces in Figure 3 (c) reflect light in a

single direction. Finally, the retro-reflective surfaces in Figure 3 (d) reflect light back in the same direction as the incoming light [64].

In computer graphics and computer vision, light specularity is an important property for creating realistic images of shiny or reflective objects. When a glossy, specular surface is illuminated, it reflects the light preferentially in a particular direction, resulting in what is known as specularity. The location of this specularity is influenced by three key factors: the position of the light source, the shape of the surface being illuminated, and the position of the viewer or camera observing the surface [64].

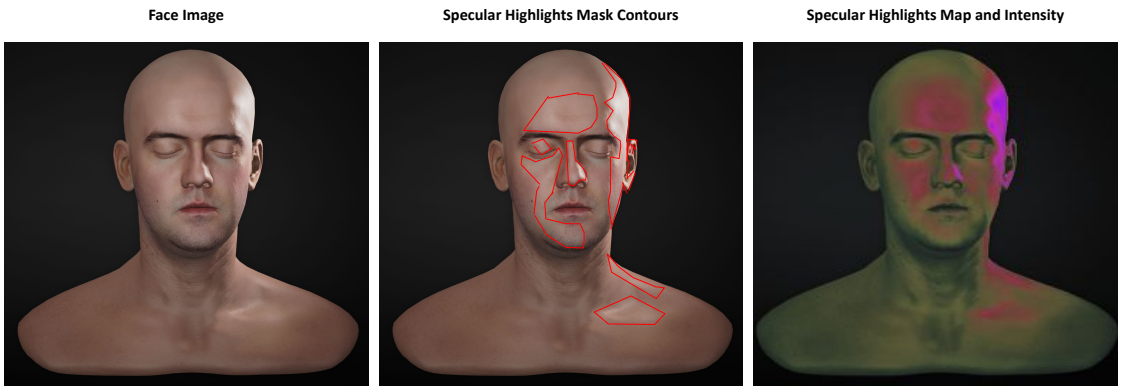


Figure 4: The specular reflection of light on human skin [8].

The specular reflection of light can be observed in a wide range of natural and artificial surfaces, such as mirrors, water surfaces, shiny metal surfaces, and human skin, as illustrated in Figure 4. The human eye's cornea is also highly specular and frequently contains a reflection of the light in the scene [64], as shown in Figure 5.

In the context of FRS, light specularity can be used to detect and prevent facial

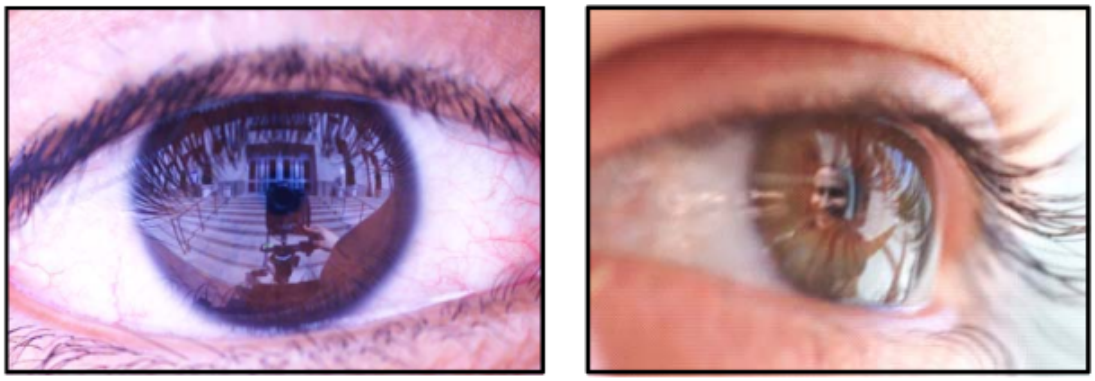


Figure 5: The corneal reflection of human eyes [134].

spoofing attacks, such as using printed or digital images. Analyzing the specular reflections on a face makes it possible to distinguish between a real face and a spoofed one, as the specular reflections on a spoofed face are usually absent or differ in their pattern from those of a real face. This technique is often used as part of a liveness detection system to prevent fraudulent access to secure systems [62].

1.4 Research Objectives and Hypothesis

The main objective of this dissertation is to enhance facial recognition security and performance. Therefore, in this dissertation, we studied and verified the hypothesis that facial spoofings struggle to fake the reflective components in their counterfeits. Thus, we developed novel countermeasures against various FRS spoofing attacks, including Deepfakes, PA, and MFDA using intelligent ML-based specular highlights detections.

1.5 Dissertation Contributions

This dissertation presents novel ML-based technologies aimed at enhancing the security of FRS against various attacks, such as identity theft, DeepFakes, liveness PA, and MFDA. To achieve this, we have developed countermeasures to identify and prevent facial biometric PA, detect DeepFakes, and identify MFDA. Our approach is based on intelligent ML-based specular highlights detections, which leverage the assumption that existing facial spoofing techniques fail to replicate reflective components accurately. The contributions of this dissertation work can be summarized as follows:

- For facial liveness detection, we design a software-based approach to enhance mobile security using a new liveness detection method named AIME, which utilizes corneal-specular reflections to detect various presentation attacks automatically. The study involved collecting two ML datasets, an eye reflection dataset to learn corneal-specular reflection detection and a corneal-specular reflection dataset to learn liveness authentication. An advanced lightweight ML package was developed to identify corneal reflective patterns and authenticate users. The proposed method was implemented as a native-like app for multiple platforms, providing a reliable multi-factor authentication solution for eye-based biometric and FRSs.

- We propose three ML-based methods, CHIEFS, MobiDeep, and READFake, to build an ensemble for DeepFake detection using reflection features from different body and facial parts instead of relying on a single feature from human eyes. New

facial image datasets are collected and annotated for corneal reflection segmentation and DeepFake detection applications. We study the impact of environmental factors, such as color and illumination conditions, on reflectance. A lightweight real-time mobile application is developed by modularizing feature extraction and embedding to address the limitations of cloud-based ML approaches. The application provides high accuracy and fast classification speed for DeepFake detection. Furthermore, we investigate various state-of-the-art DeepFake datasets and feature extractors. The proposed modular designs for feature extraction and embedding make it possible to use our tools as complementary solution modules in other existing tools.

- We design a lightweight, modular, and real-time approach named DARI to render a complementary MFDA detection module for edge and mobile FRSs. We exploit reflective elements of a human face to detect physiological flaws effectively. We generate and annotate a new DARI dataset with master and real face images for MFDA.

1.6 Dissirtation Structures

The remainder of this dissertation is organized as follows: Part 2 reviews and summarizes relevant studies on liveness detection in Chapters 2 and 3, and introduces AIME, the proposed facial liveness detection method in Chapter 4. Part 3 reviews and summarizes relevant studies on DeepFake detection in Chapters 5 and 6, and presents our novel ML-based DeepFake detection technologies, including CHIEFS, MobiDeep,

and READFake in Chapters 7, 8, and 9, respectively. Part 4 reviews and summarizes relevant research on MFDA in Chapters 10 and 11, and introduces DARI, the proposed countermeasure against MFDA in Chapter 12. Finally, Part 5 concludes the dissertation and provides recommendations for future work in Chapter 13.

PART 2

FACE LIVENESS DETECTION

CHAPTER 2

AN OVERVIEW OF FACE LIVENESS DETECTION

2.1 "It is Liveness, not Secrecy, that Counts"

Biometric authentication relies on unique physical or behavioral characteristics to identify and verify individuals. Examples of biometric identifiers include fingerprints, facial recognition, iris scans, voice recognition, and even gait analysis. While biometric authentication can be a convenient and secure way to verify a user's identity, it cannot rely solely on secrecy for its security [54].

One of the main reasons for this is that biometric identifiers are not secret. Unlike a password or PIN, which can be kept confidential, biometric information is often publicly available or can be easily obtained. For example, a person's face can be captured by surveillance cameras or found in public records. Even more concerning, biometric data can be stolen from databases that store this information, making it much more difficult to change or revoke once it has been compromised [54, 167].

Another reason why secrecy cannot be relied upon for biometric authentication is that it is vulnerable to attacks that exploit the physical nature of the identifiers. For example, an attacker could use a high-quality photograph or 3D-printed replica to trick a facial recognition system, or they could use a silicone mold to replicate a user's fingerprint. Even more advanced attacks involve creating artificial biometric data that is designed to fool a system into thinking it is a real person [167].

The biometric authentication process must include a check for liveness, which involves verifying that the biometric data is being captured from a live person and not a spoof or fake. This helps improve the security of the authentication system, even though the biometric data itself does not need to be kept secret.

2.2 Biometric Liveness Verification

Biometric liveness verification is an essential tool for protecting us against fraud and improving the security of authentication systems. Liveness verification ensures that the biometric data used to verify a user's identity is live and not a fake or altered representation. This helps prevent attacks that attempt to spoof biometric systems with photographs, videos, or other impersonation methods [149, 167].

One of the main ways that liveness verification protects us is by reducing the risk of identity theft. Biometric data is unique to each individual and cannot be easily changed or revoked once it has been compromised. This means that if an attacker were to obtain someone's biometric data, they could use it to impersonate the individual and gain access to sensitive information or areas. By using liveness verification to confirm that the biometric data is live, organizations can reduce the risk of identity theft and improve the overall security of their systems [21, 149].

Liveness verification is also essential for protecting privacy. Biometric data is highly personal, and individuals have a right to control how it is used and who has access to it. Organizations can ensure that biometric data is only used for its intended purpose by using liveness verification [21, 149].

Finally, liveness verification is vital for ensuring the accuracy and reliability of biometric authentication systems. Without liveness verification, biometric systems would be vulnerable to attacks that attempt to spoof the system with fake or altered biometric data. Therefore, organizations can improve the accuracy and reliability of their biometric authentication systems by using liveness verification to confirm that the biometric data is live and not fake [21, 149].

In summary, biometric liveness verification is a critical tool for protecting us against fraud, improving the security of authentication systems, protecting privacy, and ensuring the accuracy and reliability of biometric systems. By using liveness verification, organizations can create a more robust and secure authentication system that reduces the risk of fraud and data breaches while providing a better user experience.

2.3 Face Presentation Attack

A type of attack that aims to deceive biometric recognition is known as a Presentation Attack (PA), which is defined by ISO/IEC 30107 as an attempt to interfere with the operation of the biometric system by presenting fraudulent information to the biometric capture subsystem. When this type of attack is focused on facial biometric data, it is specifically referred to as a Face PA [11, 21].

A PA is carried out with the intention of achieving either of two objectives: impersonation or obfuscation [21]. Impersonation is when the attacker tries to use someone else's identity, while obfuscation is when the attacker tries to avoid being recognized by the biometric system.

Impersonation attacks can be carried out using different Presentation Attack Instruments (PAIs). PAIs are physical or digital objects, tools, or materials that can be used to perform presentation attacks against FRS. Digital PAIs refer to displays that can be used to reproduce real or modified digital face photos and videos, while physical PAIs are those that the human attacker can literally touch, such as silicone masks or printed masks and faces. Attackers can hold these objects in front of their faces, wear them, and attempt to trick the system into recognizing them as someone else [21].

The ISO/IEC 30107-3 standard defines three levels of spoof artifacts based on their complexity and difficulty of creation [11]:

- Level 1 (A): A level 1 (A) spoof artifact refers to a basic and easily producible PA, such as printed photos or digital images of the subject's face.

- Level 2 (B): A level 2 (B) spoof artifact is a more sophisticated PA that requires some effort and skill to create, such as preparing 2D paper masks and video displays of faces with movement and blinking.

- Level 3 (C): A level 3 (C) spoof artifact is the most advanced and difficult-to-create type of PA. It involves creating a custom-made, lifelike mask that closely resembles the target individual's facial features, including their skin texture and other physical characteristics. Creating level 3 spoof artifacts usually requires specialized equipment and expertise and is typically reserved for high-level targeted attacks.

CHAPTER 3

FACE LIVENESS DETECTION LITERATURE REVIEW

The existing face liveness detection techniques can be classified into three broad categories: hardware-based techniques, software-based techniques, and eye-based techniques. This chapter presents an overview of current liveness detection techniques and their drawbacks.

3.1 Hardware-based Techniques

Hardware-based techniques acquire the user's facial features using special add-on sensors that operate in association with a facial recognition sensor and process the captured facial data using internal software. For instance, using special sensors such as Light Field Camera (LFC) [137] or TrueDepth camera system [40] to obtain distinct facial characteristics for liveness detection. Several works also have used reflectance/multispectral properties for liveness detection [158, 173, 180]. They employed Short-Wave Infrared (SWIR) advanced imaging technology to produce images based on radiation to find how skins and masks react differently to light in order to distinguish between real facial skins and mask materials. The main limitations of hardware-based techniques are the requirements of special lighting devices, user collaboration, and expensive sensors to obtain facial liveness information.

3.2 Software-based Techniques

On the other hand, software-based liveness detection techniques do not require user cooperation or extra costly hardware components. Software-based techniques comprise hand-crafted feature extraction algorithms and deep learning features-based algorithms. The texture pattern analysis [110] is the most widely used software-based technique. The authors utilized three hand-crafted feature extraction algorithms, including Local Binary Patterns (LBP), Gabor wavelet features, and Histogram of Oriented Gradients (HOG), to obtain low-level features from facial images in order to explore the micro-textural pattern differences between real faces and PA artifacts. Although this technique is easy to implement and effective on specific databases, it can not generalize well when confronting sophisticated PAs such as hyper-realistic and 3D masks. Shao et al. [151] presented a 3D mask face anti-spoofing approach using VGG-16 deep convolutional layers to obtain dynamic facial texture features caused by facial muscle movements. Similarly, [152] used the deep dynamic texture using a joint discriminative learning model for 3D mask spoof detection. Both approaches cannot detect VR-based spoofing with dynamic facial motion patterns. Wang et al. [172] also proposed a software-based 3D face mask anti-spoofing technique by fusing texture and geometry features with facial depth information to represent the differences between real and spoofed faces. However, the proposed technique required a special camera to record depth images, which is not applicable for PAD on various mobile and IoT devices. Additionally, it is sensitive to mask qualities, and the processing requires substantial memory and computational resources.

3.3 Eye-based Techniques

Corneal-specular reflections and visual attention have been investigated in many applications, such as human-to-human interaction [97], human-computer interaction [41], human behavior tracking [157], and human activity recognition [42]. Backes et al. [36] also investigated the sensitive information that could be accessible through corneal-specular reflections. While previous studies have mainly focused on gaze and humans' daily activities tracking applications, only a few studies have concentrated on exploiting the corneal-specular reflections and eyes' visual features for liveness detection. For example, blink detection techniques [47,71] which continuously tracked the action of eyes' blinks that are performed unconsciously, or challenge-response techniques such as tracking the gaze of a user toward a predefined challenge in order to record and identify real face [32]. Another eye-based liveness detection technique [49] examined pupil dilation (pupil dynamics) to detect whether the eyes are alive or the presentation is suspicious by making the screen dark and then suddenly bright to constrict the pupil while a near-IR video sensor records a short video of the eyes. However, according to [75], this approach requires a near-IR video sensor with a special design to capture pupil dynamic dilation. A more sophisticated technique is presented in [113], which used facial images captured in a couple of sequential time points associated with the camera movement between different positions to compare the corneal-specular reflection changes. It tracked user motion using multiple sensors, including GPS, a gyroscope, and an accelerometer. However, this technique is sensitive to lighting and noise. Furthermore, it requires users' collaboration and special equipment, which is incompatible with most webcams available today.

In the following chapter, we introduce AIME, the proposed facial liveness detection method. AIME is a software-based liveness detection method, it does not require any expensive extra sensors. AIME displays a challenging pattern on the authentication screen, and then it captures the corneal-specular reflections using a front camera. Using lightweight ML, AIME processes and detects human liveness with high accuracy under subsecond time. AIME is the first work that uses human corneal-specular reflections as a challenge-response for mobile and IoT security and builds such a system with high accuracy and efficiency, to the best of our knowledge.

CHAPTER 4

APPLE IN MY EYES (AIME): LIVENESS DETECTION FOR MOBILE SECURITY USING CORNEAL SPECULAR REFLECTIONS

4.1 Background

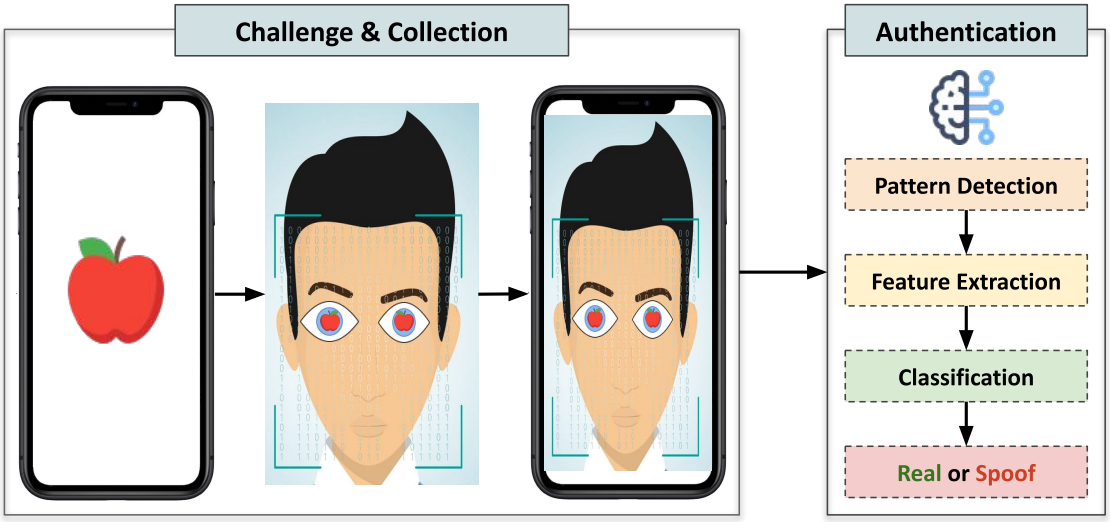


Figure 6: AIME process overview.

Biometric characteristics, including but not limited to fingerprint, voice, iris patterns, or facial features, have rapidly become popular in many mobile security applications for the applicability across multiple devices and services, automated recognition, and fast authentication experiences of individuals [117]. Additionally, the recent global pandemic caused by COVID-19 has spurred a big shift to touchless biometric authentication methods such as facial recognition due to their contactless and non-invasive process. Also, several popular touch-based authentication methods are heavily restricted for

Table 1: Difficulty levels and requirements of biometrics presentation attacks (PAs) (by Fast Identity Online (FIDO) Alliance [150]).

Type	Biometric	Difficulty	Attack Requirements
Level A	Immediate	Time: <1 day Expertise: Layman Equipment: Standard	photo printouts photo display on mobile device
Level B	Moderate	Time: <7 day Expertise: Proficient Equipment: Special	paper mask video display (with movement)
Level C	Difficult	Time: >7 day Expertise: Expert Equipment: Bespoke	3D mask theatrical mask HD video

perceptions of safety as impacts of Coronavirus [84]. Almost all modern smartphones and surveillance devices deploy facial recognition as multi-factor security for device access control, personal data security, online financial services and payment authentication. However, the vulnerability of facial recognition-based biometrics to Presentation Attacks (PA) (as known as spoofing) causes a significant challenge to their usability as they cannot assess real user physical presence in unsupervised settings. Thus liveness detection is an essential step of facial recognition-based biometrics.

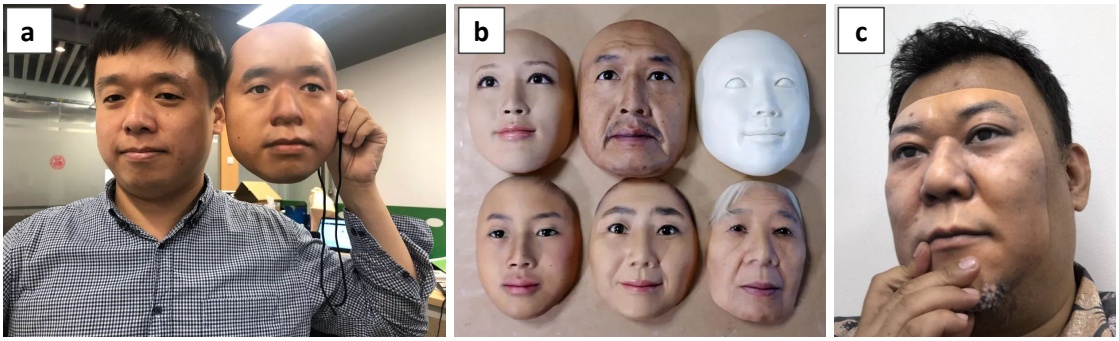


Figure 7: PA with hyper-realistic masks: a [1], b [24], and c [9].

The Fast Identity Online (FIDO) Alliance [150] has considered three levels of PA scenarios for facial recognition-based biometrics to be tested, as summarized in Table 1. Levels A and B attacks are straightforward for attackers to generate and present to biometric systems whereas Level C is more difficult to perform for attackers [138]. A Presentation Attack Detection (PAD) algorithm must detect spoofed artifacts on Levels A and B by default for a FIDO certification. However, PAD against Level C attacks, including 3D silicon masks and 3D facial models using VR, is very challenging. Several PAD techniques have been proposed to identify liveness against Level C attacks, such as hardware-based techniques, software-based techniques, and eye-based techniques. The hardware-based techniques [40, 137, 158, 173, 180] require high computational overhead, extra expensive imaging and lighting sensors (e.g., Light Field Camera (LFC) and TrueDepth camera [40]) in order to extract facial features and detect liveness accurately. The software-based techniques [110, 151, 152, 172] require high memory and computational resources and cannot overcome sophisticated attacks. The eye-based techniques [32, 47, 49, 71, 113] require users' collaboration and special equipment. In addition, they are sensitive to illumination and environmental conditions and are not useful for replay attacks. Moreover, as the quality of PA instruments (e.g., hyper-realistic masks, 3D reconstruction, and printing technologies in Figure 7) improves and the difficulty (in terms of time, expertise, equipment, and cost) of creating these artifacts decreases, achieving reliable liveness detection with the existing techniques alone remains challenging.

This chapter introduces a novel *software-based* facial liveness detection approach

named ”Apple in My Eyes (AIME),” using a challenge-response authentication protocol, AIME employs a screen display as the challenge and the valid corneal-specular reflections as the response for detecting the liveness against spoofing for mobile device security [120–122]. As presented in Figure 6, we hypothesize and validate the idea of ”Your Eyes Show What Your Eyes See! (YES2).” Therefore, AIME generates multiple security challenge patterns on the authentication screen in different sequences and captures meaningful reflective pattern responses from the user's eyes using the front camera to distinguish real humans from spoofing attacks. We design and develop various Machine Learning (ML) techniques to identify reflective patterns from the user's eyes and perform authentication, including face and eye images acquisition (Cascaded Shape Regression (CSR) [89] and Google ML Kit [5]), corneal-specular reflections detection (MobileNet-V2 Single Shot Detector (SSDLite) [146]), detected reflective pattern images super-resolution (Super-Resolution Convolutional Neural Network (SRCNN) [57]), deep learning feature extraction (VGG-16 [155], ResNet-152 [74], MobileNet-V2 [146], EfficientNet-B0 [164], DenseNet-121 [80], and Principal Component Analysis (PCA) [30]), and classification (Support Vector Machine (SVM) [147]). We compose them in a lightweight ML package to achieve liveness under a subsecond level delay (200 ms) for the entire task. We also create two ML datasets, including eye images for reflection localization and corneal reflection images for classification and learning liveness authentication. We have implemented AIME as a cross-platform application to be compatible with multiple operating systems (e.g., Android and iOS) and can, therefore, be deployed

to various mobile and IoT devices as a multi-factor authentication alternative or as a complementary software solution for eye and facial recognition-based biometrics without losing the level of security.

We have performed experiments on various devices by collecting over a thousand eye and corneal reflection images under different conditions, including daylight, dark, indoor, outdoor, wearing glasses, and gaits (laying, sitting, walking, and standing). We verify that AIME can detect levels A, B, and C attacks, including images and videos displayed on a phone or tablet screen, printed-out paper images, 3D silicon masks, and 3D facial models with real-time simulations through Virtual Reality (VR) technology. The experimental results in accuracy and performance indicate that AIME is effective and efficient in detecting the liveness against sophisticated PAs.

The main contributions of this work include:

- A new software-based liveness detection method for mobile security is introduced using corneal-specular reflections to automatically detect different presentation attacks;
- Two types of ML datasets are collected, the eye reflection dataset for learning corneal-specular reflection detection and the corneal-specular reflection dataset for learning liveness authentication;
- An advanced lightweight ML package is proposed to identify corneal reflective patterns and perform authentication;
- The proposed method is implemented as a native-like app for multiple platforms to be deployed as a multi-factor authentication solution for eye-based biometric and

facial recognition systems.

4.2 Proposed Architecture

The principal objective of AIME is to detect liveness by challenging and sensing the reflective corneal-specular patterns. AIME consists of Challenge and Pattern Detection (CPD), Feature Extraction and Classification (FEC), and Data Augmentation and Training (DAT) modules, as illustrated in Figure 25.

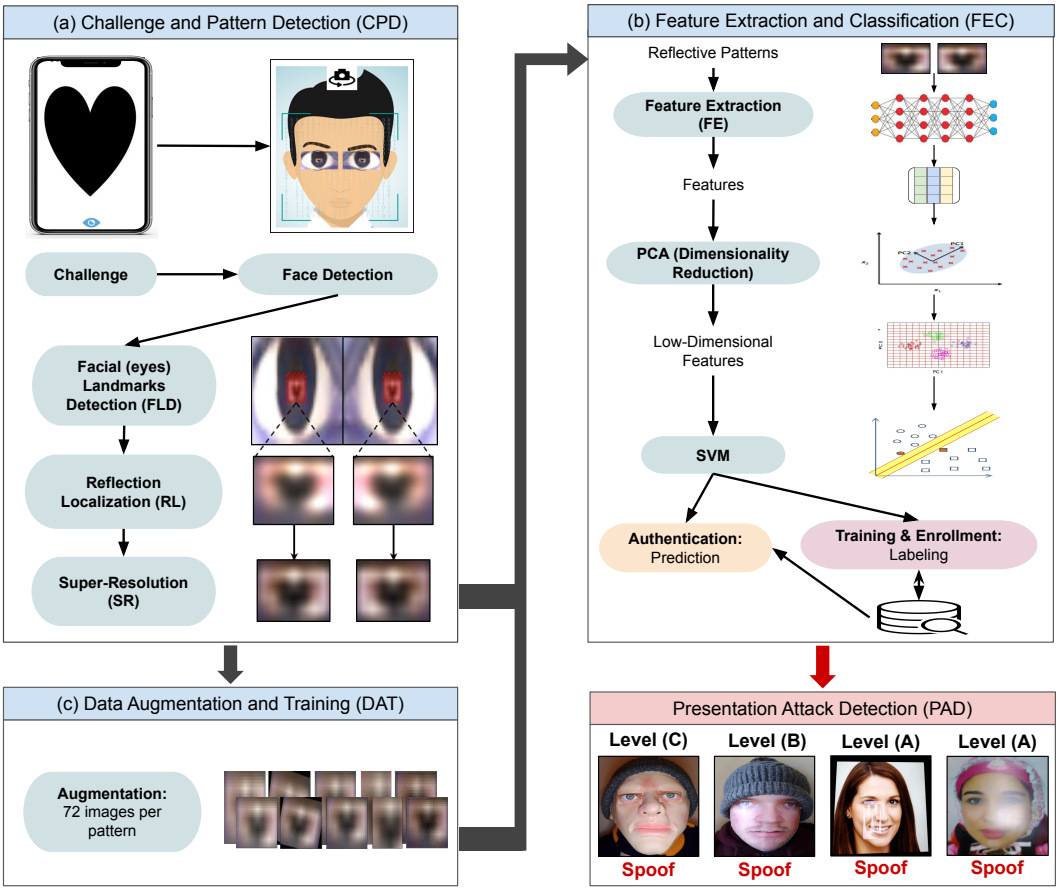


Figure 8: The block-diagram of AIME liveness detection method.

4.2.1 Challenge and Pattern Detection (CPD)

The CPD module in Figure 25 (a) performs Challenge and Face Detection (CFD), Facial Landmark Detection (FLD), Reflection Localization (RL), and Super-Resolution (SR) functions. The CFD function displays a sequence of image patterns on the authentication screen as challenges, captures facial images using the front camera, and processes captured facial images for identity verification. For authentication, we use a Cryptography Secure Pseudo-Random Number Generators (CSPRNG) algorithm [22] to generate a list of secure random numbers. It ensures that no two constant unlock attempts can obtain the same security pattern id or the same sequence of challenge patterns appearance. A pre-trained cascaded classifier [171] is used to identify faces in collected images. It is capable of processing images quickly with a high detection rate. CPD is also responsible for extracting reflective pattern images from the detected facial landmark (eyes) and generating high-resolution (HR) reflection images. FLD function in AIME is accountable for detecting right and left eyes from facial images. RL is responsible for locating the corneal-specular reflection images from the extracted eye images. We use manual annotation software called VGG Image Annotator (VIA) [61] to annotate the corneal-specular reflection area in eye images. Then, we train the RL model with the MobileNet-V2 as a feature extractor and its modified version of the Single Shot Detector (SSD), known as SSDLite. We also use the open-source TensorFlow [10] object detection API [83] to perform the task. After we extract the reflective patterns from eye images, we use Super-Resolution Convolutional Neural Network (SRCNN) [57] to recover HR images from the reflective pattern regions and enhance their perceptual quality.

4.2.2 Feature Extraction and Classification (FEC)

Using HR corneal-specular reflective pattern images extracted from the CPD module, ***the FEC module in Figure 25 (b)*** performs Feature Extraction (FE), Principal Component Analysis (PCA), and Support Vector Machine (SVM) functions.

The FE function is responsible for learning a new set of deep learning features from corneal-specular reflective pattern images. It uses transfer learning to obtain features using various pre-trained deep neural network architecture models, such as deep convolutional network (VGG-16) [155], deep residual network (ResNet-152) [74], inverted residual network with linear bottlenecking features (MobileNet-V2) [146], scaled deep convolutional network (EfficientNet-B0) [164], and densely connected convolutional network (DenseNet-121) [80]. To determine the optimal effective feature extractor model, we have evaluated AIME with different backbones for feature extraction to assess their authentication accuracy and speed.

For example, VGG-16 returns the last max-pooling layer's output by removing the final three fully connected layers from VGG-16 with an HR RGB image of 224×224 input. First, VGG-16 subtracts the mean RGB value from each pixel. After that, the input image passes through a stack of convolutional layers. Eventually, VGG-16 produces a volume shape of 25,088 ($7 \times 7 \times 512$) dimensions. When ResNet-152 deep residual network architecture is used as a backbone for feature extraction, the default input image size is 224×224 and returns a 2048-dimensional feature vector using a global average-pooling layer. Using the MobileNet-V2 as a backbone for feature extraction takes 224×224 as the default input image size. It uses the default number of filters at each layer and controls

the width of the network, defining alpha value to 1.0 (α= 1.0). MobileNet-V2 generates a 1280-dimensional feature vector using a global average-pooling layer by removing the fully-connected layer at the top of the network. When EfficientNet-B0 neural network architecture is used as a backbone for feature extraction, the default input image size is 224×224 and generates a 1280-dimensional feature vector using a global average-pooling layer. Similarly, when the densely connected convolutional network (DenseNet-121) is used as a backbone for feature extraction, DenseNet-121 takes the default input image size of 224×224 and returns a 1024-dimensional feature vector from the last global average-pooling layer. AIME FE function uses feedforward to extract features from every network and then applies PCA to reduce the dimensionality of the extracted feature vectors into the optimal lower-dimensional space (64-dimensions). SVM with Radial Basis Function (RBF) kernel [147] is used for labeling the classification at the training and enrollment stages and making predictions for authentication.

4.2.3 Data Augmentation and Training (DAT)

The DAT module in Figure 25 (c) is accountable for creating patterns, augmenting corneal-specular reflection images in different reflective conditions, and generating datasets. We first generated various *challenge image* patterns with a white background on the screen to enhance the reflection image on the cornea surface. For evaluation purposes, as shown in Figure 9, we created 15 challenge patterns, including eight black and white shapes, four black and white alphanumeric images, and three color patterns. We then built two types of ML datasets, *eye reflection data* (i.e., whole eye images that contain

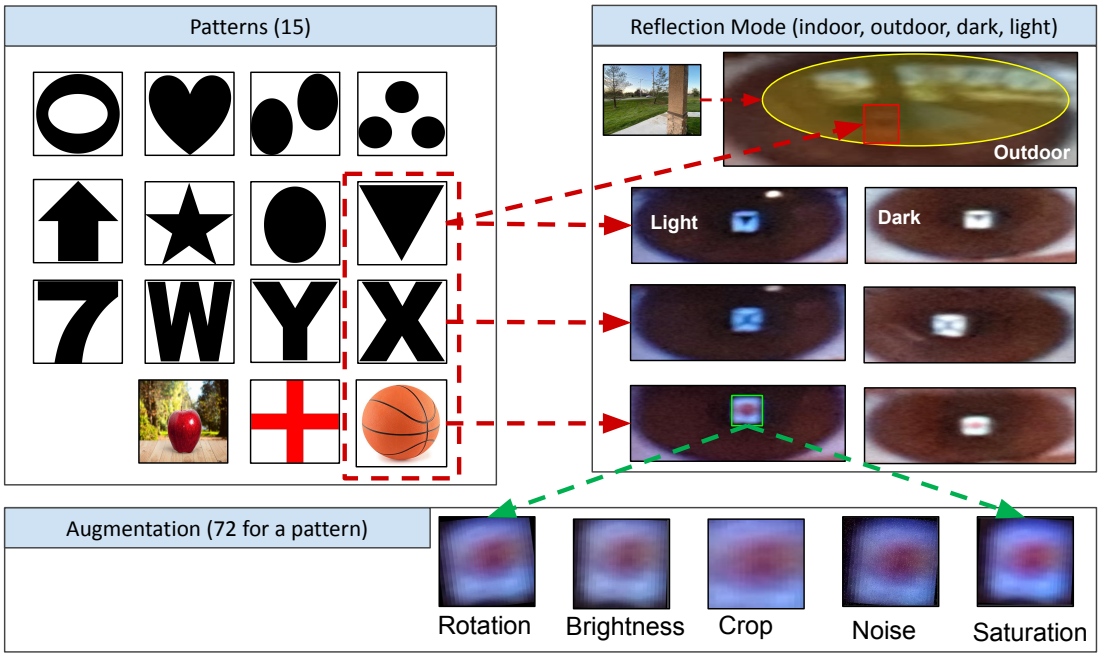


Figure 9: Reflection pattern data collection.

reflections of screen challenge patterns) and *corneal reflection data* (i.e., the reflection of challenge patterns on eyes) for learning liveness authentication. We have collected eye reflection images from diverse environments, including daylight, darkness, indoor, outdoor, wearing glasses, and various postures (lying, sitting, walking, and standing). We have used the AIME application to collect facial images using different challenge patterns, including shapes, letters, numbers, and colors. We have built a dataset of 1,300 eye reflection images for training the reflection location learning. We have created a dataset of 1080 reflection pattern images for the initial training by augmenting 72 images from every 15 patterns for classification in SVM. We have used the image data augmentation technique to create modified versions of the enrollment images to improve the classification model's performance. We augment corneal-specular reflection images using rotation,

crop, noise, saturation, and brightness variations.

4.3 Evaluations

We carried out extensive experiments using AIME cross-platform implementation in Android, iOS, and web to evaluate its performance under real-world scenarios. We primarily used the AIME mobile application on Samsung Galaxy S9 (SM-G960F) and iPhone 11 to check its authentication delay and accuracy. SM-G960F screen's size is 5.8 inches. It also comes with a 2.7 GHz Octa-Core processor, 64 GB memory, 4 GB RAM, and a 3000 mAh battery. In addition, it has dual selfie cameras with resolutions of 8MP and up to 30 frames per second imaging rate for the primary camera and 2MP for the dedicated iris scanner camera. iPhone 11 comes with a screen of size 6.1 inches, an A13 Bionic chip (with 6-core CPU, 4-core GPU, and 8-core Neural Engine), 128 GB memory, 4 GB RAM, and a built-in rechargeable lithium-ion battery. It has a TrueDepth front camera with 12MP resolution and an extended dynamic range for video at 30 fps, 4K video up to 60 fps, 1080p HD video at 60 fps, and auto image stabilization. We also used the AIME web application to collect corneal-specular reflection images from the 15 patterns using Microsoft LifeCam HD-3000, which supports the resolution of 1280 by 720 pixels for video and 1280 by 800 pixels for a static image. Authentication delay is measured as the interval between when the authentication system detects the facial authentication event and when the system generates the result, including the time for data collection, reflection detection and processing, feature extraction, and classification. We set the subject-camera distance between 20 cm and 50 cm during the data collection stage.

AIME temporarily increases its screen brightness for better reflection.

4.3.1 Performance of Reflection Detection

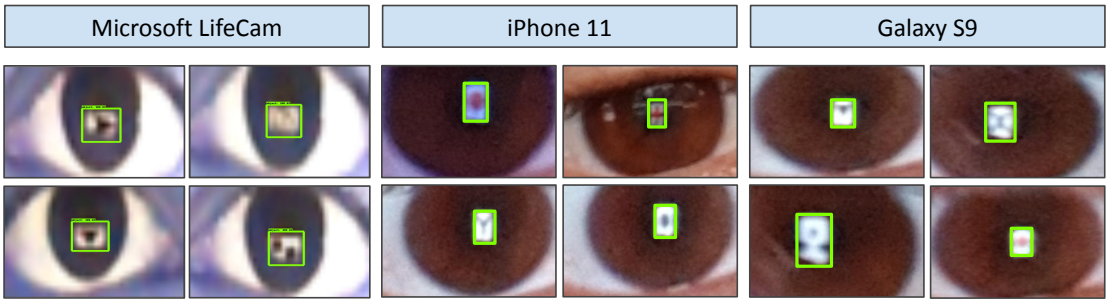


Figure 10: Samples of reflection detection results.

Table 2: The reflection detector results (%) on the testing datasets.

iPhone 11		Galaxy S9		Microsoft LifeCam	
mAP	mAR	mAP	mAR	mAP	mAR
87.36%	88.92%	87.28%	88.38%	87.10%	89.74%

AIME uses a reflection detection model (i.e., reflection localization) to identify the corneal-specular reflection presence (classification task) and its location (localization task) within eye images. The reflection detection model predicts a bounding box that surrounds the corneal-specular reflection. It then assigns a class (screen reflection) and a confidence score to the predicted box. We primarily evaluated the reflection detection model using mean Average Precision (mAP) and mean Average Recall (mAR) to quantify how the model performed across images in our test set and at various confidence thresholds and object size variations.

We trained MobileNet-V2 to detect corneal-specular reflections in three datasets.

The iPhone 11 image dataset consists of 434 eye images collected by the iPhone 11 app. We split the dataset in an 80:20 ratio. 80% (347 images) of the data were used for training the model, while 20% (87 images) were used for validation and testing. As presented in Table 2, the reflection detection model achieves 87.36% mAP and 88.92% mAR. We also trained MobileNet-V2 to detect the corneal-specular reflection in the Galaxy S9 eye images dataset, consisting of 433 eye images. We split the dataset into 80% (346 images) for training the model and 20% (87 images) for validation and testing. The reflection detector model obtains 87.28% mAP and 88.38% mAR. Also, we trained MobileNet-V2 to detect the corneal-specular reflection in the Microsoft LifeCam eye images dataset, consisting of 433 eye images collected using Microsoft LifeCam HD-3000 installed on desktop and laptop with 21 and 13-inch monitors. We split the dataset in an 80:20 ratio as well. The reflection detector model scores 87.10% mAP and 89.74% mAR. Figure 10 shows the reflection detection sample results using iPhone 11, Galaxy S9, and Microsoft LifeCam testing datasets. Although Microsoft LifeCam's reflection locations are slightly bigger than others due to the larger challenge screen size, all devices can detect explicit reflection images.

4.3.2 Performance of Authentication with Different Backbone Models for Feature Extraction

The primary goal of the experiments is to assess the feasibility of AIME usage on mobile devices by checking the performance of AIME authentication accuracy and speed on both Android and iOS. In addition, we also conducted an ablation study using various deep neural network architectures, including VGG-16, ResNet-152, MobileNet-V2,

EfficientNet-B0, and DenseNet-121, as backbones for feature extraction to find suitable feature extractor models for the mobile application.

Table 3: A comparison of CPD's average delay on Android and iOS.

Type	CPD Delay (ms)
Android	203.00
iPhone	212.59

Table 4: A comparison of authentication performance using different backbone models for feature extraction on Android and iOS.

Backbones	Type	FEC Delay (ms)	FEC Acc
AIME (DenseNet-121)	Android	71.36	96.00%
	iPhone	75.61	
AIME (EfficientNet-B0)	Android	50.12	97.00%
	iPhone	58.13	
AIME (MobileNet-V2)	Android	43.76	97.00%
	iPhone	50.41	
AIME (ResNet-152)	Android	64.82	98.00%
	iPhone	68.31	
AIME (VGG-16)	**Android**	**56.70**	**99.99%**
	iPhone	**59.52**	

As shown in Tables 3 and 4, we performed authentication delay experiments using a react-native based mobile application with the TensorFlow.js react-native package on both Samsung Galaxy S9 (SM-G960F) and iPhone 11. We collected the average delay of CPD and FEC processes, respectively. Besides, as presented in Table 4, we used five deep neural network architectures (VGG-16, ResNet-152, MobileNet-V2, EfficientNet-B0, and DenseNet-121) for feature extraction. Using transfer learning to obtain deep learning features from reflective pattern images, we reused layers from previously trained

models on the ImageNet dataset [52] and removed the fully-connected layer at the top of every network except VGG-16, we removed the final three fully connected layers. Then, PCA was used to reduce the dimensionality of the extracted feature vector from every network into lower-dimensional space (64-dimensions). Finally, we trained the SVM classifier in the FEC module using the obtained lower-dimensional feature vectors. Five SVM classifiers were trained using the corneal reflection training dataset (870 images) and tested on the corneal reflection testing dataset (210 images).

Tables 3 and 4 show that CPD average delay, FEC average delay, and SVM classifiers accuracy with different feature extractors. Overall, AIME is highly effective, it achieves over (95%) accuracy in classifying corneal reflection pattern images by using various types of screen patterns (i.e., 15 patterns), and its average authentication delay is less than (300 ms), which presents an insignificant performance overhead. AIME (VGG-16) classifier's accuracy is better than (99.9%) with an average authentication delay of (259.70 ms) on Samsung Galaxy S9 and (272.11 ms) on iPhone 11. AIME (ResNet-152) is the second-best in accuracy (98%), and its average authentication delays are (267.82 ms) and (280.90 ms) on Samsung Galaxy S9 and iPhone 11, respectively. AIME (MobileNet-V2)'s accuracy is the third-best (97%), and it has the lowest average authentication delay on both Samsung Galaxy S9 (246.76 ms) and iPhone 11 (263 ms). AIME (EfficientNet-B0)'s classifier also scores (97%) accuracy and achieves the second-lowest average authentication delay of (253.12 ms) on Samsung Galaxy S9 and (270.72 ms) on iPhone 11. AIME (DenseNet-121)'s accuracy is the least (96%), and its average authentication delay is the highest (274.36 ms) on Samsung Galaxy S9 and (288.20 ms) on iPhone

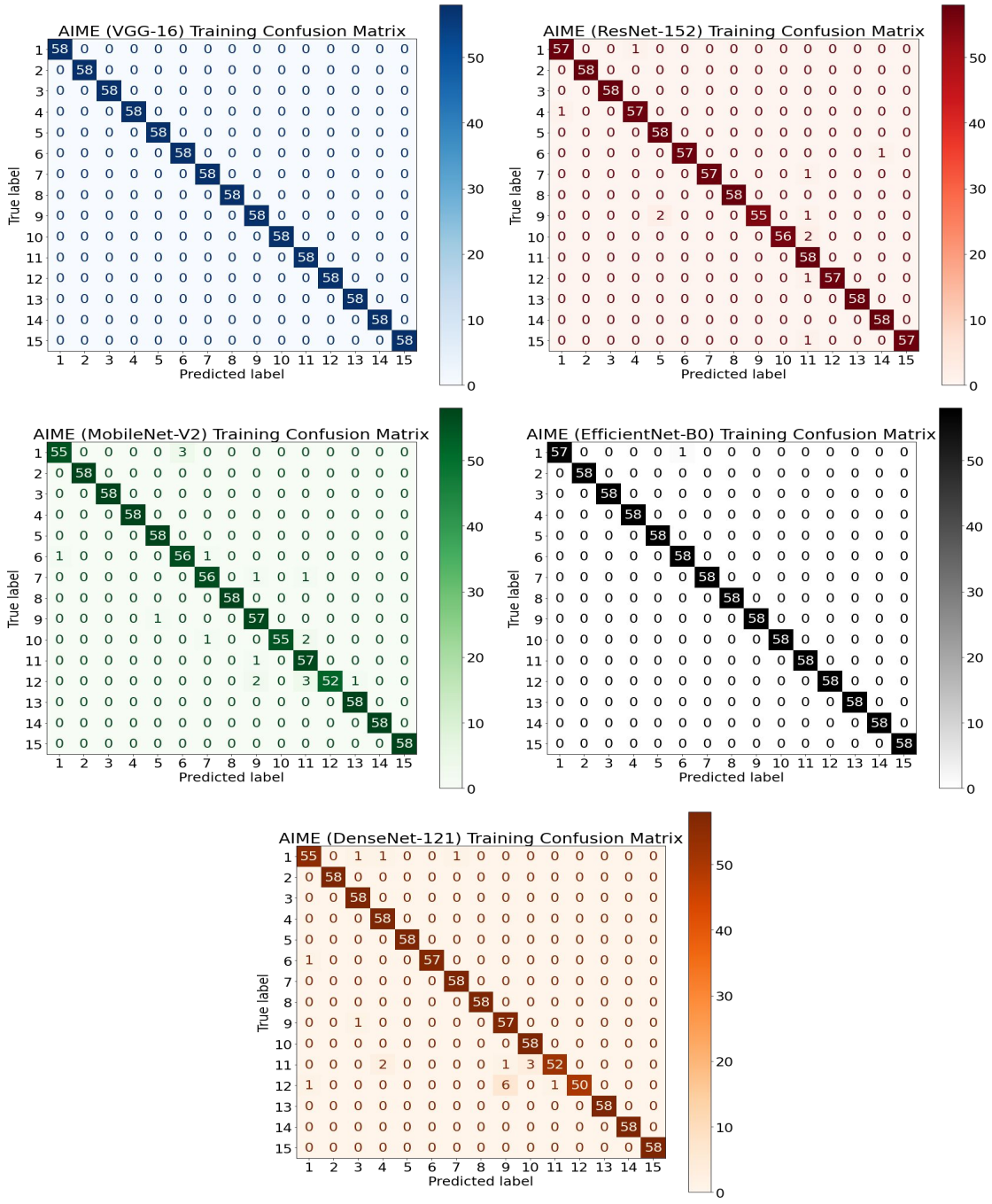


Figure 11: AIME SVM classifiers' prediction confusion matrices for corneal reflection training dataset using different backbone models for feature extraction.

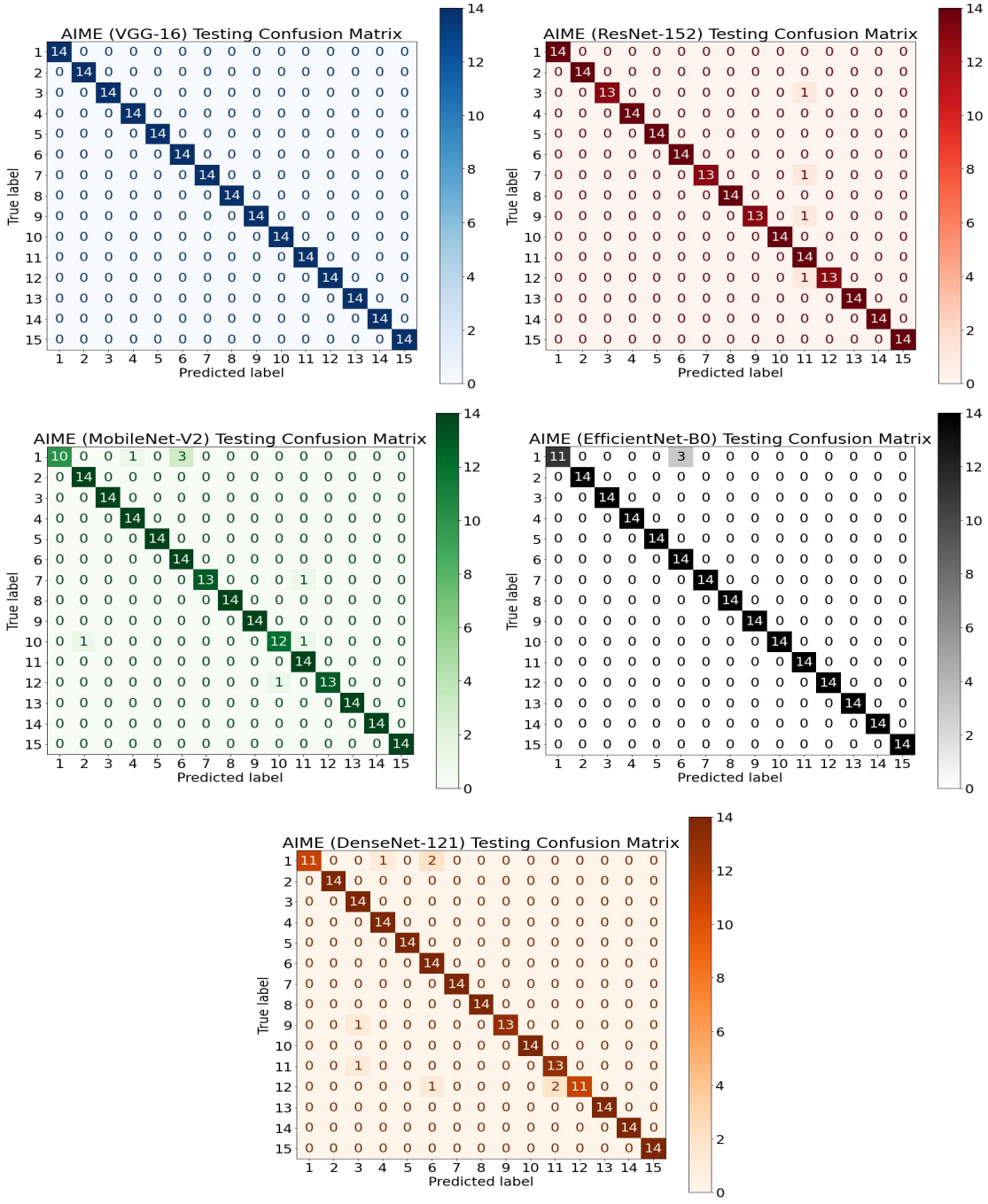


Figure 12: AIME SVM classifiers' prediction confusion matrices for corneal reflection testing dataset using different backbone models for feature extraction.

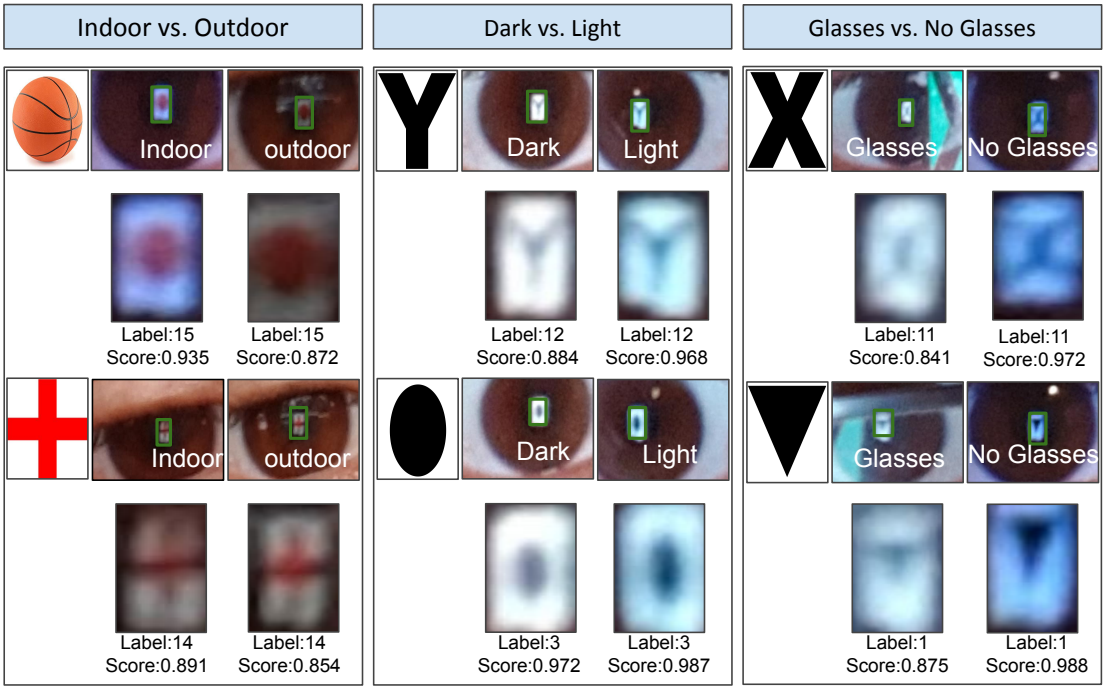


Figure 13: SVM classifier performance on the corneal reflection testing dataset.

11. To quantify the accuracy of our SVM classifiers, the confusion matrices in Figures 11 and 12 show the training and testing error rate. For instance, the top left-hand panels of Figures 11 and 12 show the training and testing error rate of AIME (VGG-16), all corneal reflective pattern images in our training and testing datasets are classified correctly by AIME (VGG-16) SVM model.

Figure 13 shows some SVM classifier performance samples from the testing dataset. We tested the AIME classifier with the images collected from four different modes (dark, light, indoor, and outdoor). Besides, we tested reflections with and without glasses. Figure 13 shows six challenge patterns (1, 3, 11, 12, 14, and 15) and predictions of images collected from different illumination conditions. For example, in challenge patterns 15

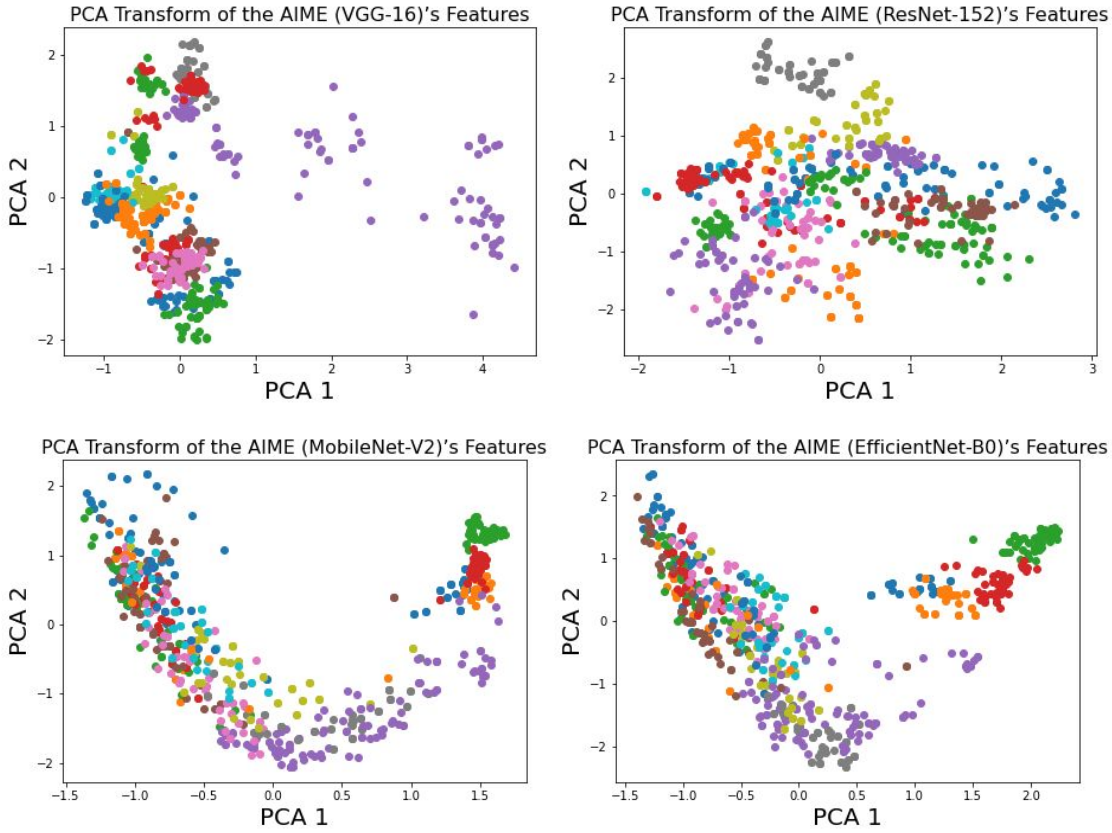


Figure 14: Principal Components Analysis (PCA) transform of the training dataset features using different backbone models for feature extraction. Top left: AIME (VGG-16)'s features. Top right: AIME (ResNet-152)'s features. Bottom left: AIME (MobileNet-V2)'s features. Bottom right: AIME (EfficientNet-B0)'s features.

and 14, we compare reflective pattern classification in indoor and outdoor modes. Our model correctly predicts outdoor reflective pattern labels with noise reflected from the scene. For challenge patterns 12 and 3, Figure 13 shows testing images in dark and light modes. For challenge patterns 11 and 1, we show a comparison of patterns classification with and without glasses. As shown in the top left-hand panel and the top right-hand panel of Figure 14, there is clear evidence that corneal reflective pattern images with the

same type tend to be located near each other in these two-dimensional representations which are obtained using AIME (VGG-16) and AIME (ResNet-152). In addition, the result of the AIME five SVM classifiers indicates that the combination of AIME deep learning feature extractor backbones and PCA features proved to be an effective method in corneal reflection pattern recognition. It also implies that AIME can detect liveness in different environments and illumination conditions. Furthermore, according to Android Profiler [18], AIME one-time power usage is trivial (i.e., 17.3mw on average), and there is no power drain during the normal smartphone operation.

4.3.3 Evaluation of Presentation Attack Detection Scenarios

According to FIDO recommendations [11], we have evaluated AIME with three levels of Presentation Attack Instruments (PAI) scenarios in Table 1. We focus on the effectiveness of the AIME PAD method alone without combining other face recognition modules (e.g., turning off Face ID in iPhone 11 and Face Unlock in Samsung Galaxy S9).

4.3.3.1 Level A (Immediate) Attacks

are easy to deploy as many biometric sources such as photos (both online and offline) are available from social media. To test a level A attack against AIME, we prepared paper printouts and tablet displays of face images from the Flickr Faces HQ (FFHQ) dataset [87]. We printed six different face samples on different surface-type papers (matte and glossy). For the digital photo presented on a tablet screen, we prepared a few sample images and displayed them on the ASUS Chromebook Flip C214, which has a standing screen display size of 11.6 inches. We use both Samsung Galaxy S9 (SM-G960F) and

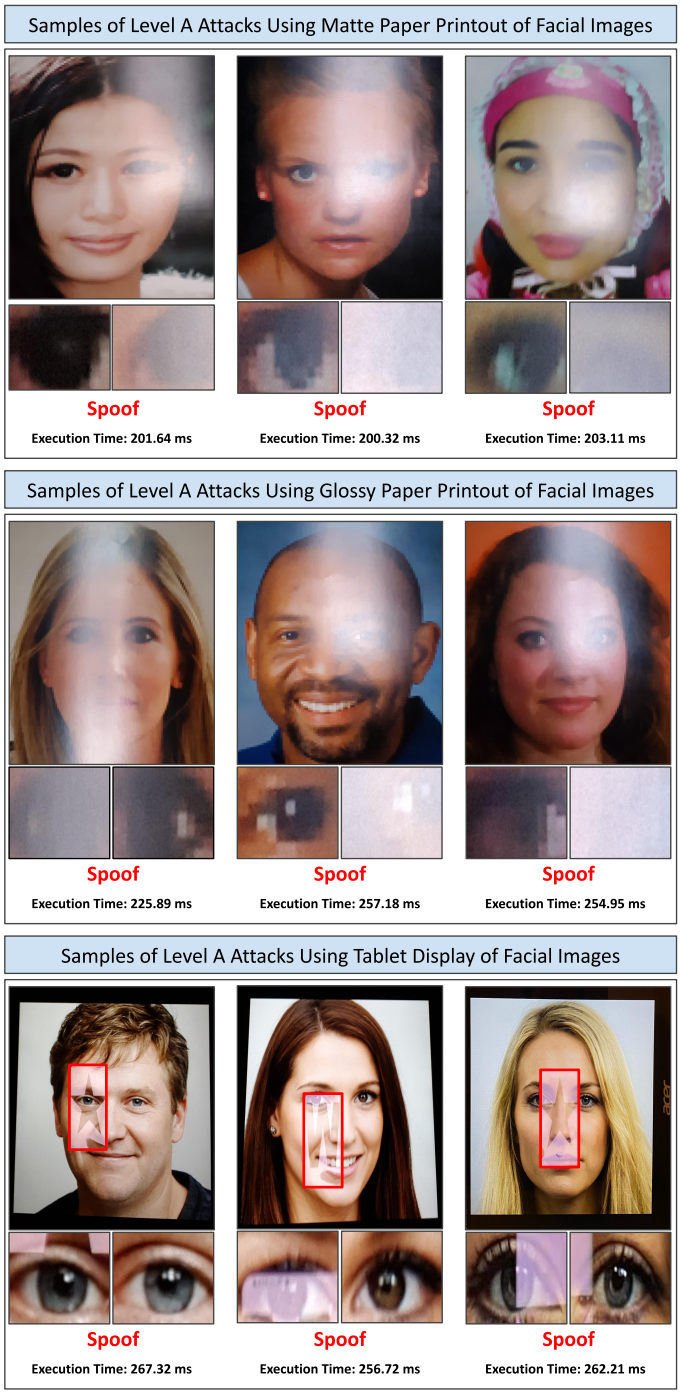


Figure 15: Samples of level A (immediate) attacks, including tablet display, glossy and matte paper printouts.

iPhone 11 cameras for the level A scenarios. Figure 15 shows that AIME detects all level A attacks as a spoof within about 270 ms due to the deformation and lack of complete reflection images around the corneal area.

4.3.3.2 Level B (Moderate) Attacks

require moderate skills, special equipment, and more time to prepare. To perform the level B attacks, we printed two URME paper surveillance masks [25] on heavy card stock papers. We altered some of the masks by making small holes (e.g., mask (2)) and wearing glasses. We use both Samsung Galaxy S9 (SM-G960F) and iPhone 11 for testing. As shown in Figure 16, the covered eyes in the mask (1) are classified as a spoof (an accurate classification) due to the lack of corneal reflection for the challenging pattern. Even if glasses can reflect some image patterns, it is still classified as a spoof because the reflection images are not in the right reflection location. Also, the reflective patterns do not match the challenging pattern. It is interesting to see the results of the paper mask (2). Even if it has small eye holes, the reflections are unacceptable because the mask becomes an obstacle blocking screen light from reaching the cornea and distorting the reflections.

4.3.3.3 Level C (Difficult) Attacks

are the most difficult attacks requiring expert skills, bespoke equipment, and a longer time to prepare. For example, Virtual Reality (VR) based spoofing attacks require advanced skills to obtain biometric characteristics. As shown in Figure 17, we reconstructed a virtual 3D facial model from a single image via [85], then we used the same approach explained in [176] to present the 3D facial model using VR system. The virtual

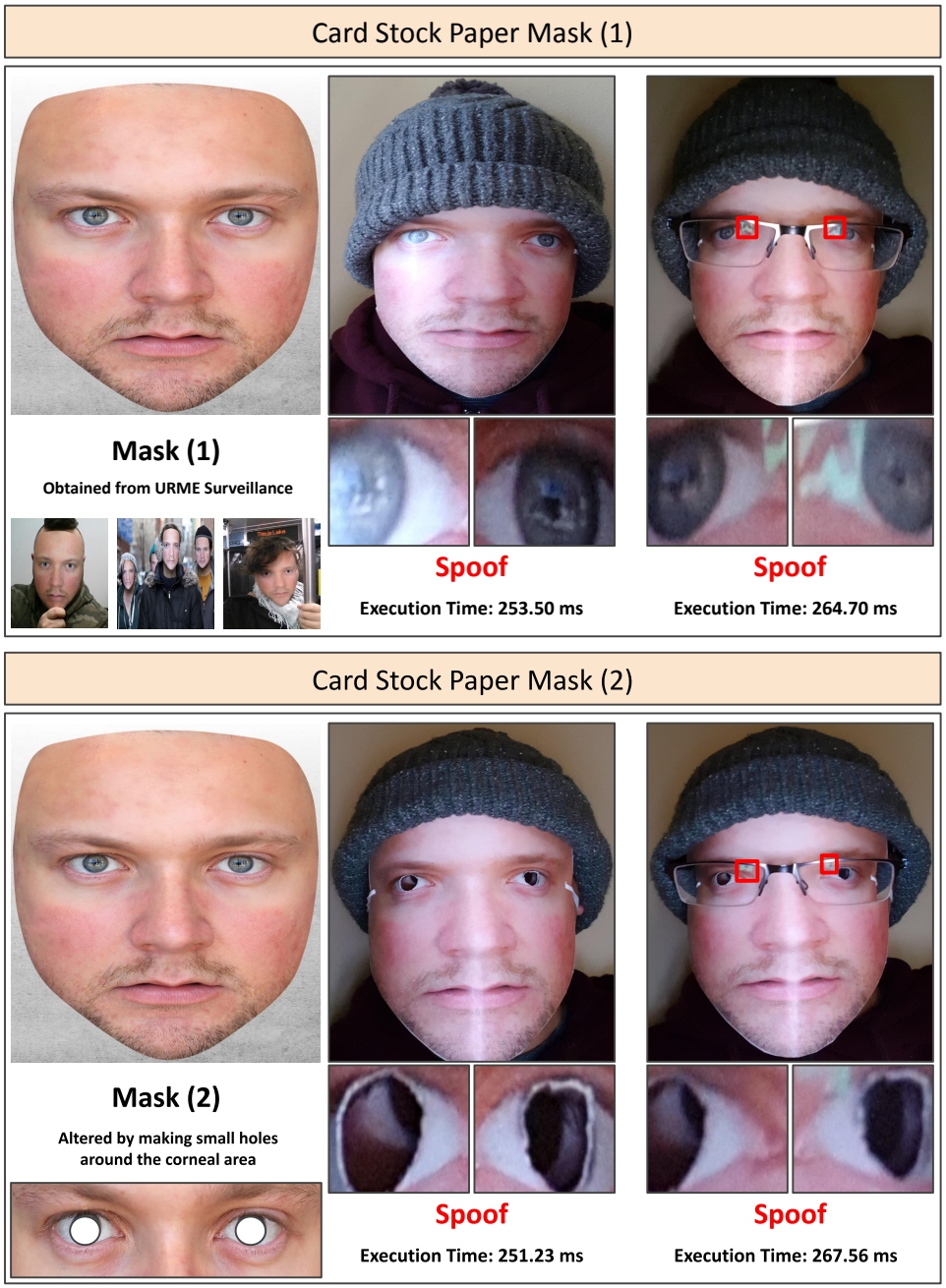


Figure 16: Samples of level B (moderate) attacks using 2D paper masks.

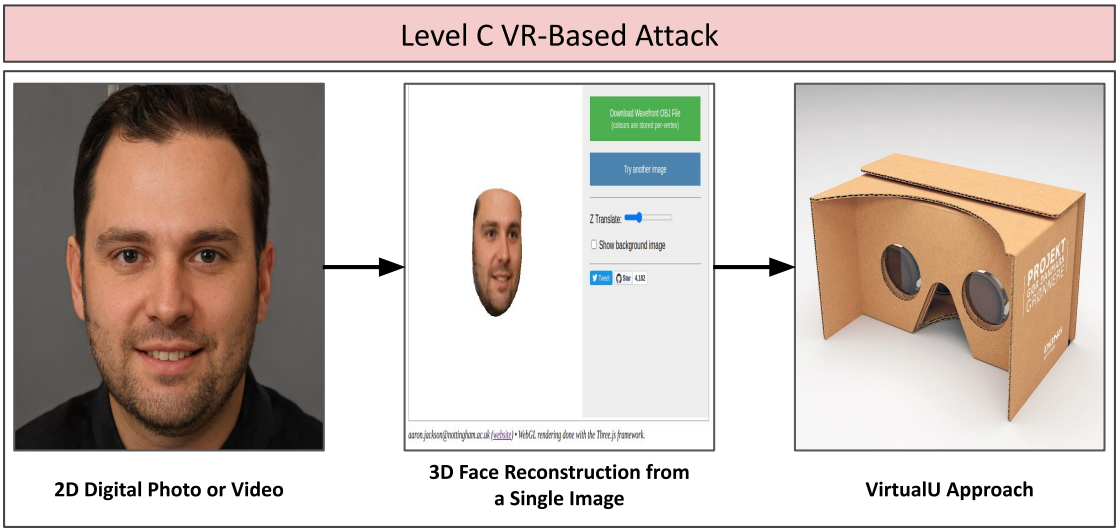


Figure 17: VR-based spoofing attack.

3D facial model of the user is displayed on the screen of the VR device with a real-time 3D simulation. Therefore, when the device rotates and translates, the 3D face moves accordingly. AIME is resilient against VR-based spoofing attacks since the VR device prevents screen challenge pattern reflection. We also evaluate the level C attacks against AIME using realistic silicone masks. As presented in Figure 18, we prepared two silicone masks, one with eye holes and wearing glasses and the second with fake eye images covering the eye holes. The level C attacks are correctly predicted as a spoof due to the lack of correct reflective images around the corneal area. AIME can prevent realistic silicone masks regardless of the eye covers. Even if the eye is open to the faker's natural eyes, AIME can detect spoofing if face recognition is used with iris detection.

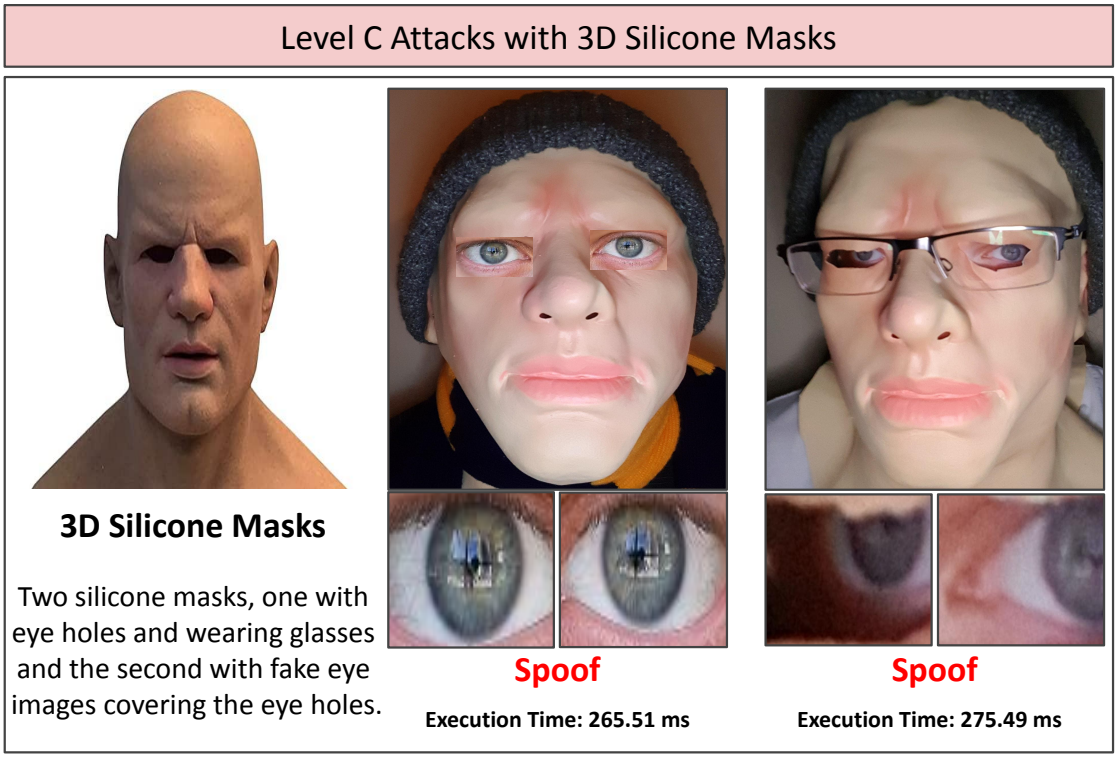


Figure 18: Level C attacks with 3D silicone masks.

4.3.4 A Comparison of Software-based PAD Methods

This section presents a comprehensive comparative analysis of different software-based liveness detection methods based on multiple criteria (e.g., technique, type of features, classifier, accuracy, application environment, hardware and processing requirements, applications, and PAD scenarios). Table 5 summarizes the comprehensive comparison of software-based liveness detection methods.

A passive face spoofing detection method using the micro-texture analysis was presented in [110]. The authors used three hand-crafted feature extraction algorithms to obtain features from facial images, including Local Binary Patterns (LBP), Gabor wavelet

features, and Histogram of Oriented Gradients (HOG). They transformed the extracted features vector into compact linear representations and then used linear SVM for classification. Their method was tested using images of real subjects and high-quality photo attacks recorded in different environmental and illumination conditions using conventional webcams with a 640×480 pixels resolution. Their method achieved 99.99% accuracy in detecting levels A and B attacks. However, the proposed method is less efficient in detecting level C attack scenarios and requires high computation.

Wang et al. [172] also proposed a passive 3D face mask anti-spoofing method by fusing texture and shape features. This method rebuilt facial depth information from RGB images through 3D Morphable Model (3DMM) and obtained texture and geometry feature sets using the LBP hand-crafted algorithm and deep learning model to represent the differences between real and spoof faces. For classification, they used SVM and deep neural network models. This method was tested using videos of real faces and mask attacks recorded by a Microsoft Kinect sensor under controlled conditions. The proposed method achieved 100% accuracy in detecting 3D mask attacks. However, it requires a special camera to record depth images, such as the Microsoft Kinect sensor, which is not applicable for liveness detection on various mobile and IoT devices. Furthermore, the processing requires considerable memory and high computational resources, which is consequently power-consuming.

Shao et al. [151] presented a passive 3D mask face anti-spoofing approach using VGG-16 deep convolutional layers to obtain dynamic facial texture features caused by

facial muscle movements, such as eye blinking and lip movements. Similarly, [152] exploited the deep dynamic texture using a joint discriminative learning model for 3D mask spoof detection. Both [151] and [152] tested their approach using real faces and masked faces videos recorded through the Microsoft Kinect sensor and Logitech C920 web camera with the resolution of 1280×720. Furthermore, both approaches selected one frame in every five consecutive frames in each input video to extract dynamic information representation, which resulted in a 51-frame sequence. The authors in [151] used SVM classifier and achieved 98.93% average accuracy. Likewise, [152] used SVM classifier and achieved 97.44% average accuracy on various datasets. However, both approaches may not be able to distinguish such subtle differences in dynamic textures of the VR-based spoofings with various facial motion patterns and real faces. Besides, it requires substantial memory and computational resources to process 51 frames for authentication.

In contrast, AIME uses a lightweight ML-based technique and successfully detects levels A, B, and C sophisticated attacks such as 3D masks and VR-based spoofings using deep learning features obtained from corneal reflective pattern images. Moreover, AIME can identify reflective patterns and perform authentication in different environmental and illumination conditions with high accuracy at around 200 ms using only the front camera.

4.3.5 Discussion

The primary purpose of this study is to investigate the feasibility of using corneal-specular reflections for liveness detection in mobile and IoT devices. Our approach uses screen display and human corneal-specular reflections as a challenge-response method

Table 5: A comprehensive comparison of software-based PAD methods.

PAD Methods	**Micro-texture Analysis Method [110]**	**Shape-based Method [172]**	**Facial-motion Estimation Method [151]**	**Dynamic-texture Method [152]**	**Ours (AIME)**
Technique	Passive	Passive	Passive	Passive	Passive
Features	LBP, Gabor Wavelets, and Histogram of Oriented Gradients (HOG)	Texture featurs (LBP), shape features (3D Morphable Model), and deep learning features	Deep convolutional dynamic texture features	Facial-motion features (optical flows estimation) and deep convolutional dynamic texture features	Deep learning features
Classifier	SVM	SVM and deep neural network	SVM	SVM	SVM
Accuracy	99.99%	100%	98.93%	97.44%	99.99%
Application Environment	Indoor and outdoor	Controlled environment	Controlled environment	Controlled environment	Indoor and outdoor
Hardware Requirement	Conventional webcam with resolution of (640 $\times$ 480 pixels)	Microsoft Kinect	Microsoft Kinect and Logeitech C920 webcam with resolution of (1280 $\times$ 720 pixels)	Microsoft Kinect and Logeitech C920 webcam with resolution of (1280 $\times$ 720 pixels)	Camera with minimum resolution of (1280 $\times$ 800 pixels) and screen
Software Processing Requirement	High	High	High	High	Low
Applications	Facial recognition systems	Facial recognition systems	Facial recognition systems	Facial recognition systems	Facial and eye-based biometric systems
PAD Scenarios	Paper printouts	Paper printouts	Paper printouts	Paper printouts	Paper printouts
	Photo displays	Photo displays	Photo displays	Photo displays	Photo displays
	2D paper masks	2D paper masks	2D paper masks	2D paper masks	2D paper masks
	Video displays	Video displays	Video displays	Video displays	Video displays
	-	3D masks spoofing	3D masks spoofing	3D masks spoofing	3D masks spoofing
	-	-	-	-	VR-based spoofing

by creating multiple screen patterns as a challenge, then detecting corneal-specular reflections as a response using a front camera and analyzing the reflective pattern images using lightweight ML techniques. In summary, the findings from the experimental results include the following: AIME can detect and return corneal-specular reflection images of various screen patterns with different color and size variations. AIME achieves very high accuracy in classifying corneal-specular reflection pattern images using various deep neural network architectures as backbones for feature extraction, including VGG-16 (99.99%), ResNet-152 (98.00%), MobileNet-V2 (97.00%), EfficientNet-B0 (97.00%), and DenseNet-121 (96.00%). AIME accurately performs liveness detection in different environments with different illumination conditions on multiple mobile devices at around 200 ms against various types of sophisticated PAs, such as paper printouts, photo display, 2D paper mask, video display, 3D mask spoofing, and VR-based spoofing. AIME requires the user's attention (open eyes and look at the screen) in order to authenticate. However, it can still authenticate users passively, assuming they are holding the device in front of their faces. Moreover, AIME's lightweight ML package does not require significant computation overhead or costly extra sensors, making it fully integrable with other contactless biometric authentications.

4.4 Summary

AIME is a software-based human liveness detection method, proposing and applying the notion of "Your Eyes Show What Your Eyes See! (YES2)" for mobile device security. As a challenge-response method for liveness of mobile authentication, AIME

uses screen display and human corneal-specular reflection. We designed and built multiple Machine Learning (ML) functions to identify reflective patterns and perform authentication, including eye image acquisition, reflection image augmentation, super-resolution, feature extraction, and classification. We have also created a couple of ML datasets (e.g., eye images for reflection localization and corneal reflection images for super-resolution and classification) for learning liveness authentication. We have built a lightweight ML package for Android, iOS, and web applications. We have demonstrated that AIME provides an accurate and efficient PAD using only a front-facing camera, without using any infrared or depth sensors, through extensive experiments under diverse conditions. AIME has a broad applicability, as it can be used either as a stand-alone human liveness detection app or for various mobile and IoT device apps as a complementary software solution for touchless biometric systems.

PART 3

HUMAN VISUAL DEEPFAKE DETECTION

CHAPTER 5

AN OVERVIEW OF DEEPFAKE

5.1 DeepFakes: The Threat to Trustworthy Visual Information

The rise of a new generation of digitally manipulated media has caught the attention of researchers, policy-makers, and the public [16, 23]. Artificial intelligence (AI) advances have enabled the production of highly convincing fake videos that depict people doing or saying things they have never done or said before. These fabrications are commonly referred to as ”deepfakes,” a term coined from ”deep learning” and ”fake” [118].

DeepFakes can be defined as manipulated or synthetic media that seem authentic and feature individuals who appear to say or do something they have never said or done. They are produced using AI techniques, including machine learning and deep learning [118].

The rapid development in the production and manipulation of synthetic media has raised concerns about potential misuse. DeepFakes can be used to extort, harass, humiliate, or blackmail victims, leaving individuals, companies, and government institutions at risk [26].

The adverse effects of DeepFakes and synthetic media go beyond the immediate harm caused by their misuse. It leads to a loss of public trust in digital content as they can never be sure if what they see is manipulated or not. This scenario poses a significant challenge to democracy and national security, as DeepFakes could be used to spread

misinformation and undermine public trust in information [23, 26].

Recently, there has been a lot of focus on synthetic media, especially DeepFakes, due to the rapid advancements in technology. These advancements are making it increasingly difficult to distinguish between real and fake content [23]. For example, Nightingale et al. [132] evaluated the photorealism of AI-synthesized faces. The results indicate that synthesis models are capable of creating faces that are indistinguishable and more trustworthy than real faces. Additionally, the technology used to create DeepFakes is becoming more affordable, accessible, and easy to use [23].

5.2 Trends in Deepfake and Synthetic Media Technologies

Several factors have contributed to the emergence of various trends in DeepFake and synthetic media technologies, such as [23]:

- Access to datasets, high-quality algorithms, pre-trained models, and computing power: The computer vision community has developed large datasets of facial images to train machine learning algorithms like Generative Adversarial Networks (GANs) and Autoencoders, which can analyze a set of images and generate new ones with high quality. Additionally, pre-trained models are shared among deepfake creators, eliminating the need for training models on datasets and computing power. Furthermore, cloud computing services have made it possible to train machine learning algorithms at a much lower cost than before, making it easier to create high-quality DeepFake. Even a regular computer or smartphone with internet access can now achieve this purpose.

- The latest advancements in mobile connectivity networks: 5G connectivity offers increased bandwidth, enabling users to stream and view high-quality video content, transforming the media landscape to evolve in the direction of user-generated content.

- The availability of 3D sensors in the latest generation of consumer devices: 3D sensors can capture 3D information of entire scenes and scan objects. This can be beneficial for the creators of DeepFakes as they can obtain 3D data to generate high-quality media.

- Increased image forensics and DeepFake detection capabilities: The use of DeepFake detection technologies helps to enhance the quality of DeepFake. By providing feedback to the algorithms used to create DeepFakes, detectors assist in improving their learning capacity.

Therefore, these factors lead to several trends in DeepFake and synthetic media technologies [23]. For instance, with the advent of 5G and cloud computing, users are now able to manipulate video streams in real-time. This opens up possibilities for the application of Deepfake technologies in video conferencing, live-streaming services, and television [23]. In addition, the growing demand for AI-generated media has led to the creation of supply and demand platforms for manipulated content. These marketplaces allow users to request DeepFake videos and images [7, 17, 136]. Furthermore, the easy access to advanced computing and algorithms has caused a rise in DeepFake technology availability. These technologies are easily accessible and come with guides, making them

easy to use for those with technical expertise [7]. There are also user-friendly smartphone apps available that require no technical knowledge [20, 28]. Even chatbots on platforms like Telegram have been known to produce DeepFake images, including ones that exploit women and minors [23]. Moreover, the utilization of photorealistic 3D avatar technology in combination with AI-based deepfake technology holds significant potential for various applications. This powerful combination can enable the creation of highly realistic and lifelike virtual representations of individuals, which can be utilized for a wide range of purposes, including entertainment and online education and training [125]. Finally, DeepFake creators are now focusing on developing algorithms that can generate high-quality output with minimal input. For example, some algorithms can generate videos based on a single picture of the target or produce audio speeches resembling the target's voice based on only a few seconds of audio. This means that large amounts of visual data of a specific person are no longer necessary, making anyone with only a small number of audio-visual representations on the internet a potential target [23].

CHAPTER 6

DEEPFAKE DETECTION LITERATURE REVIEW

In this chapter, first, we discuss DeepFake creation techniques. Next, we briefly present the current DeepFake detection datasets available. Then, we review notable related DeepFake image and video detection techniques and their limitations.

6.1 DeepFake Creation Techniques

The existing human visual DeepFake creation techniques can be classified into four categories: Face reenactment, face replacement, face editing, and face synthesis DeepFakes [118]. Face reenactment DeepFake [67, 96, 136] uses a source person image to drive the target person's expression, mouth, gaze, pose (head position), or body (whole-body pose). With face reenactment, an attacker can impersonate the target's identity and control what the target can say or do. A face replacement DeepFake [7, 100, 133] aims to replace a target person's facial landmarks with that of a source person, preserving the source person's identity. Common types of face replacement attacks are face transfer and face swap. A face editing DeepFake is where a target person's attributes are added, changed, or removed, such as modifying the target's appearance by changing his or her weight, ethnicity, or age. Finally, face synthesis DeepFake [86, 88] is where a DeepFake image (fake identity) is generated without a target person as a basis. Using editing

and synthesis techniques, an attacker can build a fake identity to mislead other individuals [118].

6.2 DeepFake Detection Datasets

We summarize DeepFake detection datasets available in Table 6. Generations of DeepFake detection datasets are classified based on the number of frames and videos, as presented in [56].

6.3 DeepFake Detection Techniques

There are several notable studies that proposed techniques for DeepFake detection, using various machine learning architectures and models. The existing DeepFake detection techniques comprise deep learning-based, physical-based, and physiological-based techniques.

6.3.1 Deep Learning-based Techniques

The deep learning-based DeepFake detection techniques trained DNN to learn deep hierarchical features and the classifiers jointly in an end-to-end manner in order to identify fake faces from real ones. For example, Do et al. [55] used VGG-16 neural network architecture with pre-train weights of VGG-Face to detect DeepFake faces. [43, 99] utilized luminance and chrominance color components to improve DeepFake detection. Mansourifar et al. [111] applied one-shot learning to determine out-of-context objects that appeared in DeepFake to distinguish synthesized faces from real ones. [112] employed incremental learning for DeepFake detection. Guo et al. [69] proposed an attention-based

approach to detect DeepFakes using inconsistency of the right and left eyes. Authors of [124] proposed a hybrid combination of supervised and deep reinforcement learning to improve the generalization of DeepFake detectors. [58] proposed an identity consistency transformer DeepFake detection technique using identity consistency in the image. [143] designed a hybrid model for video DeepFake detection using a combination of CNN and RNN architectures to extract temporal optical flow features from video frames. In [78], authors proposed a frame inference-based detection framework to learn the referenced representations of the video frames and predict the future frame's representations using an autoregressive model and representation prediction loss. [77] proposed a pairwise learning model to distinguish the features between fake and authentic images. [181] created multi-attentional network architecture to capture local discriminative features from multiple face regions. Nguyen et al. [126] also developed a multi-task DeepFake images detection approach which performed classification and segmentation using an autoencoder model containing an encoder and a Y-shaped decoder.

6.3.2 Physical-based Techniques

The physical-based DeepFake detection techniques concentrated on identifying artifacts and inconsistencies between the face and the physical world, such as using illumination and reflection. For instance, Matern et al. [115] detected GAN-synthesised faces through corneal specular reflection, which are either missing or appear simplified as a white blob.

Hu et al. [79] proposed a DeepFake detection technique that used the inconsistency of the corneal specular highlights between the two synthesized eyes, assuming that two eyes look at the same scene, their corneal specular highlights should show high similarities. This technique can distinguish between the real and GAN-synthesized faces when light sources are visible to both eyes and the eyes are distant from the light source. However, when these two conditions are defied, [79] will raise many false positives. [51] examined the spatial, temporal, and spectral consistency of eyes and gazes in five domains (e.g., visual domain, geometric domain, temporal domain, etc.) to classify DeepFake videos.

6.3.3 Physiological-based Techniques

The physiological-based DeepFake detection techniques utilized the physiological characteristics of real human faces and obvious artifacts in generated faces, such as asymmetric faces [178], facial geometric information [161], abnormalities in the configuration of facial landmarks [178], eyes' inconsistent iris color [115], and irregular pupil shapes [68], to identify DeepFakes. However, these techniques can not generalize well when confronting highly realistic DeepFake because they only consider single artifacts of eyes, such as iris color, pupil shapes, or similarity of corneal reflections on both eyes. In addition, such artifacts may not always be available due to the limitations of the images with blurriness, low-quality images, or occlusions.

6.3.4 DeepFake Detection on Mobile Devices

The Korea Advanced Institute of Science and Technology (KAIST) recently proposed a cloud-based mobile application service called KaiCatch [13] for DeepFake detection. However, KaiCatch requires users to download the KaiCatch mobile application and register the service to upload the images or videos to be tasted. After three to four days, a classification result (fake or real) will be sent back to the user, and for more detailed results sent via email, it charges $1.76 per image. Nowadays, anyone can make realistic DeepFake media using easy-to-use DeepFake creation mobile applications such as Reface, Avatarify, or Wombo. Using generated DeepFake media for defamation, blackmailing, and harming innocent individuals' credibility, necessitates having a real-time DeepFake detection mobile application to quickly and precisely identify forged media.

To address the limitations of the existing DeepFake detection techniques, our proposed techniques, including CHIEFS, MobiDeep, READFake, and DARI, are designed to efficiently detect sophisticated DeepFakes using specular reflection highlights from various body parts (e.g., eyes, nose, cheeks, etc.) along with the environmental factors, since most DeepFake creation techniques can not coordinate their fakes with the reflective components and the surrounding illumination and environmental conditions. Therefore, unlike most existing DeepFake detection techniques, our techniques detect various features from the specular reflection highlights, such as color components, shapes, and textures, and check the coordination with the surrounding environmental factors such as indoor/outdoor, bright/dark, backgrounds, and light strength.

Table 6: DeepFake detection datasets [56].

Datasets	Size	Identity	Methods	Quality
1st Generation DeepFake Detection Datasets				
DF-TIMIT [94]	960 videos (640 fake and 320 real)	43 subjects	Replacement	low
UADFV [178]	98 videos (49 fake and 49 real)	49 subjects	Replacement	low
FaceForensics++ [142]	5000 videos (4000 fake and 1000 real)	-	Replacement, Reenactment, Editing & Synthesis	low
2nd Generation DeepFake Detection Datasets				
Google-DF [60]	3000 fake videos	28 subjects	-	low
Celeb-DF [103]	6229 videos (5639 fake and 590 real)	59 subjects	Replacement, Editing & Synthesis	low
3rd Generation DeepFake Detection Datasets				
DFFD [50]	299039 images (240336 fake and 58703 real) and 4000 videos (3000 fake and 1000 real)	-	Replacement, Reenactment, Editing & Synthesis	low & high
DFDC [56]	128154 videos (104500 fake)	960 subjects	Replacement, Editing & Synthesis	high

CHAPTER 7

CHIEFS: CORNEAL-SPECULAR HIGHLIGHTS IMAGING FOR ENHANCING FAKE-FACE SPOTTER

7.1 Background

The AI-fueled production and manipulation techniques of fictitious human facial images, DeepFake, have accomplished notable advancement. Due to the sophisticated DeepFake generation technologies [88], [95], [136], it is getting harder to distinguish the forged images by eye. Despite many benign applications such as fun memes, visual effects, and realistic avatars, the generated fake media can be malignantly used by spreading misinformation on social media, creating deception for identity theft, and causing manipulation on election security. Hence, DeepFake has become a pandemic risk to authenticity, privacy, and security for our society. DeepFake detection technologies have become essential vaccines to mitigate the possible malignant risks.

There has been a large number of research works to detect DeepFakes. For example, [175] proposed an attention-based DeepFake detection distiller by applying frequency domain learning and optimal transport theory in knowledge distillation to improve the detection of low-quality DeepFake images. Le et al. [98] explored the asynchronous frequency spectra of color channels to train unsupervised and supervised learning models to identify GAN-based synthetic facial images. [166] extracted deep features from facial images using a Convolutional Neural Network (CNN). Another technique [102] checked eye

blinking motions, which tended to be missing in DeepFake videos using the Long-Term Recurrent Convolutional Network (LRCN). Sun et al. [161] also detected DeepFake using facial geometric characteristics. However, previous methods lacked detection generalization on unseen data because they were trained on datasets containing few low-quality video frames generated with a single model and fewer subjects. In addition, eye-based DeepFake detection techniques in [68], [79], [102], and [115] only focused on a single artifact of eyes, either iris color, blinks, or similarity of corneal reflections on both eyes. Hence, they failed to detect sophisticated DeepFake media.

This chapter presents a novel ML-based DeepFake detection technology named **CHIEFS** (**C**orneal-Specular **H**ighlights **I**maging for **E**nhancing **F**ake-Face **S**potter) [119]. As shown in Figure 24, we focus on the most reflective area of a human face, eyes, upon the hypothesis that DeepFake technologies, such as replacement and synthesis, are hard to coordinate their counterfeits with the reflective components. We seek similarity and consistency of corneal-specular highlights (CSH) with multiple surrounding semantics, such as illumination and environmental conditions that are hard to forge. Thus, instead of checking a single aspect of the eyes, we extract multiple features, including *CSH*s' color components, shapes, and textures. In addition, we extract facial images surrounding environmental factors (SEF) to check the ensemble of the reflectance with the *SEF* such as indoor/outdoor, bright/dark, backgrounds, and light strength. CHIEFS embeds the *SEF* into the feature extraction and classification process to detect the symmetricity and consistency in both eyes' color components and reflection patterns.

As illustrated in Figure 20, CHIEFS consists of a couple of ML components,

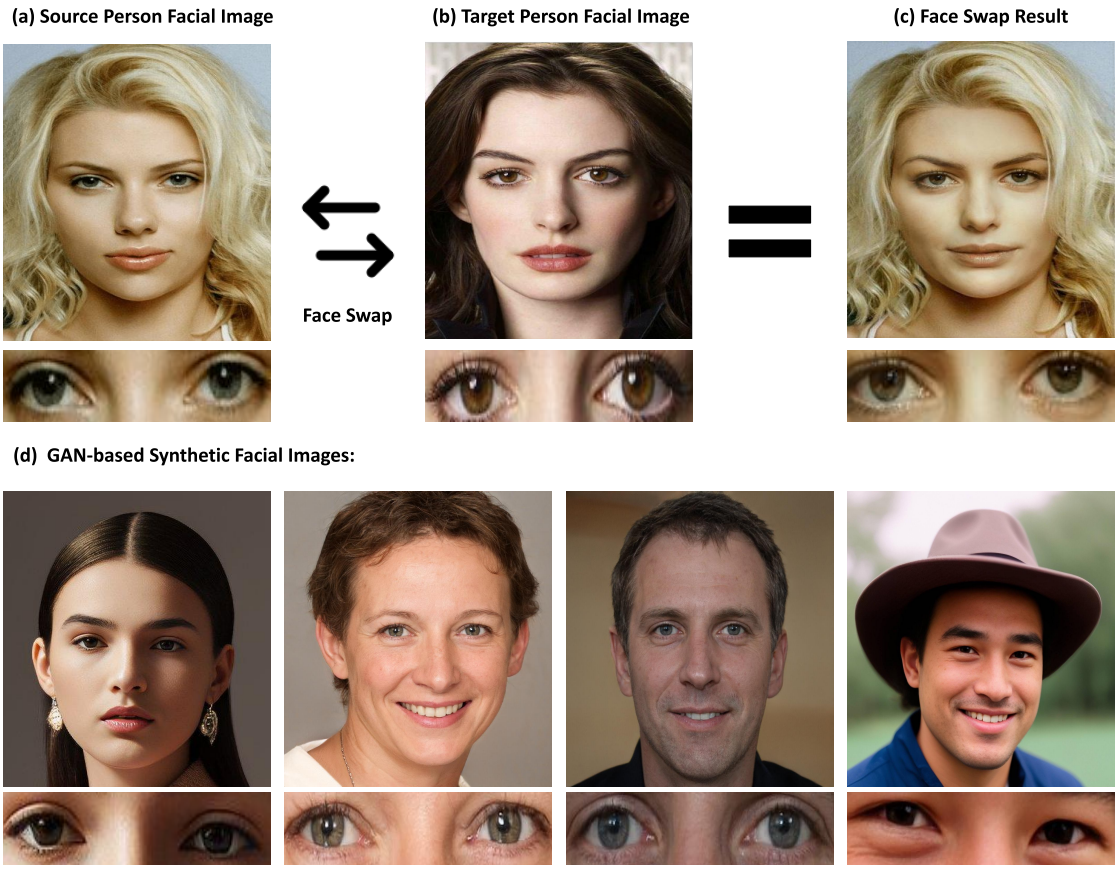


Figure 19: Samples of real and DeepFake facial images with their reflective elements. (a) and (b) are both real, (c) is a DeepFake face generated using the Face Swapper online tool [6], and facial images in (d) are GAN-based synthetic faces from [88] and [14].

including Training Data Collection and Annotation (TDCA), Highlights and Environmental Factors Detection (HEFD), and Feature Extraction, Embedding, and Classification (FEEC). The TDCA involves creating and annotating a new dataset named CHIEFS DeepFake Detection (CHIEFS-DFD). The CHIEFS-DFD dataset includes real and GAN-generated DeepFake facial images annotated with various *CSH* and environmental information. The HEFD detects right and left *CSH*, as well as identifies the *SEF* features. The FEEC extracts features from the *CSH* images, measures the right and left corneal highlights consistency (CHC), embeds additional *SEF* features, and classifies the input facial images as fake or real. We use Siamese Convolutional Neural Networks (SCNN) with various configurable neural network backbones, including ResNet-50-V2 [74], VGG-16 [155], Xception [45], and DenseNet-201 [80], for the feature extraction. We have conducted experiments with various GAN-generated DeepFake datasets to validate the accuracy of CHIEFS. The results show that CHIEFS achieves 99.00% accuracy in detecting highly realistic DeepFake facial images. Further, the modular design of CHIEFS renders itself as a complementary DeepFake detection module for any existing tools to limit the potential harm from DeepFake.

The main contributions of this work include:

- A new facial images dataset is collected and annotated for corneal reflection segmentation and DeepFake detection applications.
- A ML method is proposed to build an ensemble with various facial reflection features instead of a single feature.

- We study the impact of environmental factors on reflectance by collecting various parameters such as color and illumination conditions.

- We made modular designs for feature extraction and embedding to make it portable to other existing tools as a complementary solution module.

7.2 Proposed Architecture

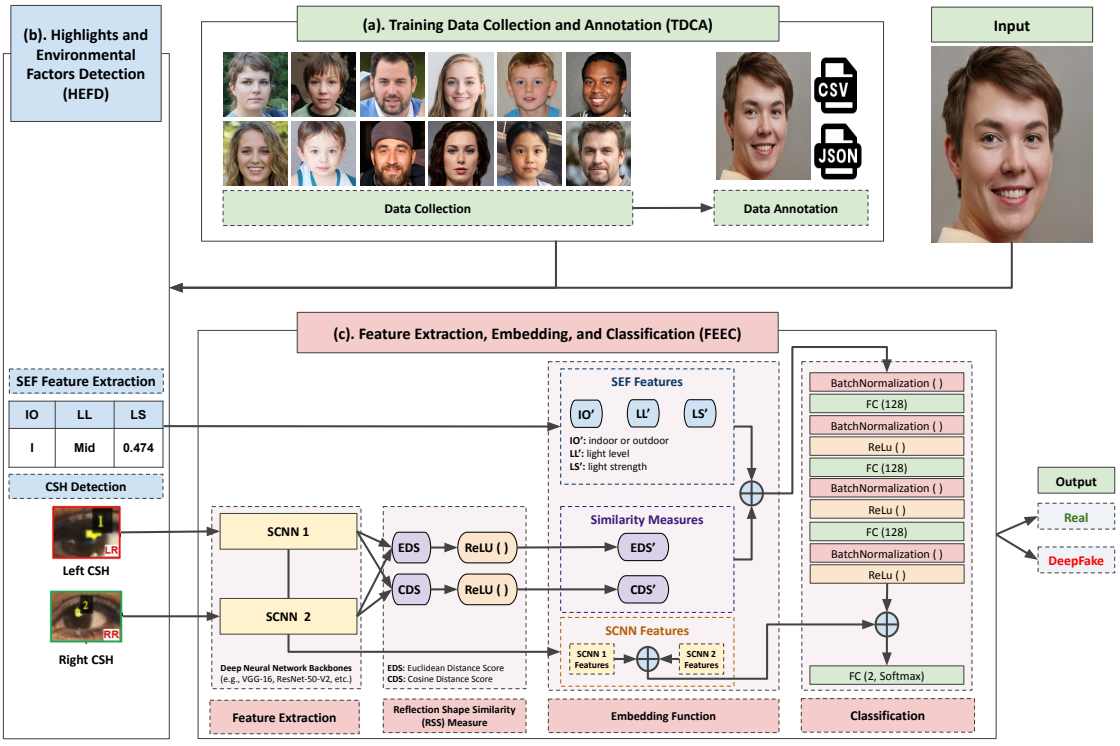


Figure 20: The CHIEFS architecture block-diagram.

CHIEFS is an ML-based DeepFake detection technology that analyzes facial images' corneal-specular highlights consistency (CHC) and checks the ensemble of the highlights with multiple surrounding environmental factors (SEF). CHIEFS is designed in a hierarchical structure, and its components are separated into three modules. Training Data Collection and Annotation (TDCA), Highlights and Environmental Factors Detection (HEFD), and Feature Extraction, Embedding, and Classification (FEEC) modules in Figure 20. The modular structure of CHIEFS allows agile updates of every module, like adding new features and enhancements according to specific use cases, as well as making CHIEFS available as a complementary DeepFake detection module for other existing tools.

7.2.1 Training Data Collection and Annotation (TDCA)

Current DeepFake detection datasets, such as UADFV [178], FaceForensics++ [142], Celeb-DF [103], and DFDC [56] do not contain the *CSH* annotation or facial image environmental factors information. Therefore, the main responsibility of ***the TDCA module in Figure 20 (a)*** is to create CHIEFS-DFD dataset [3] by collecting and annotating real and GAN-generated DeepFake facial images. We manually label the right and left *CSH* and provide the facial image-specific *SEF* information using the VGG Image Annotator (VIA) software [61]. The CHIEFS-DFD dataset contains 1,285 facial images in high resolution. 716 real facial images were collected from different datasets, including Flickr Faces HQ (FFHQ) dataset [87], Celeb-DF dataset, FaceForensics++ dataset,

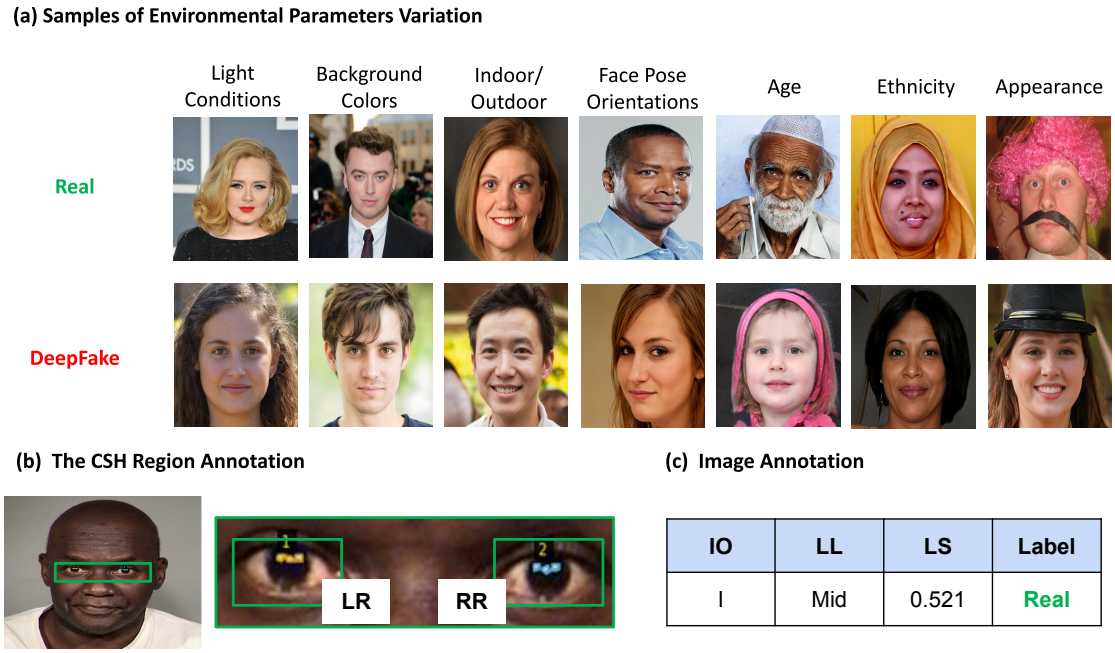


Figure 21: Environmental parameter samples and annotations in CHIEFS-DFD dataset.

and DFDC dataset. Additionally, 569 GAN-generated DeepFake facial images were acquired from various DeepFake detection datasets and human visual DeepFake generation tools, such as StyleGAN2 [88], StyleGAN3 [86], FSGAN [133], DeepFaceLab [136], and FaceShifter [100].

As illustrated in Figure 30 (a), the CHIEFS-DFD dataset contains DeepFake and real facial images in high resolutions with different environmental parameters, including illumination conditions, background colors, indoor or outdoor settings, face pose orientations, age, ethnicity, and appearances (e.g., wearing makeup and accessories). As demonstrated in Figure 30 (b) and Figure 30 (c), the CHIEFS-DFD-dataset contains two types of annotations. The *CSH* region annotation in Figure 30 (b) defines the shapes and locations

of *CSH* and classifies them into right-reflection and left-reflection classes. The *Image Annotation* in Figure 30 (c) identifies the image label (either Real or DeepFake), along with *SEF*, including indoor or outdoor (IO), light level (LL), and light strength (LS). The CHIEFS-DFD dataset contains the 2,570 annotated *CSH* segmentation masks for 1,285 facial images (two eyes per facial image). In addition, 959 images (74.63%) are labeled as indoor, and 362 images (28.17%) are labeled as outdoor. Furthermore, collecting and analyzing the distribution of CHIEFS-DFD dataset facial images' *LS* values (explained in Subsection 7.2.2) results in different LL classes (806 mid images (62.72%), 258 low images (20.07%), and 221 high images (17.19%)).

7.2.2 Highlights and Environmental Factors Detection (HEFD)

The HEFD module in Figure 20 (b) performs two major tasks, including *SEF* feature extraction and *CSH* detection. The *SEF* parameters include *IO*, *LS*, and *LL*. We train a MobileNet-V2 model on the Dense Indoor and Outdoor Depth (DIODE) dataset [169] and labeled facial images from the CHIEFS-DFD dataset (total 20,420 images) to classify the *IO* of an input image. To calculate the *LS*, we convert the input image's color space into a LAB format. The *L* channel is independent of color information in the LAB color space and only encodes intensity. The other two channels *A* and *B* encode color. Then, we extract the *L* channel and normalize it by dividing all pixel values by the maximum pixel value to have an *LS* value of the input image. Using the *LS* value, we identify an *LL* into the low, mid, and high classes (e.g., according to the *LS* distribution, the *LL* is a low if *LS* is less than 0.419, high if *LS* is greater than 0.637, and a mid if it

is in between). To detect the right and left reflections, we train the *CSH* detection model using the MobileNetV2-SSDLite [146] to detect the bounding boxes of right and left *CSH* regions and class labels.

7.2.3 Feature Extraction, Embedding, and Classification (FEEC)

Using the right and left *CSH* images and the *SEF* extracted from the HEFD module (7.2.2), ***the FEEC module in Figure 20 (c)*** performs four primary functions, including deep hierarchical feature extraction using Siamese Convolutional Neural Network (SCNN) model with configurable neural network backbones, reflection shape similarity (RSS) measure, similarity measures (*RSS*), environmental factors (*SEF*), and CSH features embedding, and classification.

7.2.3.1 Feature Extraction:

As shown in Figure 20 (c), two SCNN models with the same weights and network architecture receive the right and left *CSH* images in parallel. Various configurable neural network backbones can be used for feature extraction, including VGG-16, Xception, ResNet-50-V2, and DenseNet-201. The two SCNN models use feedforwards to extract features using a global max-pooling layer by removing the fully-connected layer at the top of every network ($include_top$= False). We do not need activation and classes because we only use the backbone models for feature extraction. Then, we use the right and left *CSH* features to measure *RSS* using euclidean and cosine distance scores.

7.2.3.2 Reflection Shape Similarity (RSS) Measure:

CSH can be detected in various shapes, which can be deformed in different colors according to illumination conditions and blended into the background. Furthermore, *CSH* can be occluded by glasses, eyelids, or eyelashes, and only a tiny portion of the reflection can be visible. Hence, the similarity measures of a single factor, such as the shape or color of the *CSH* alone, cannot be a strong indicator for classifying DeepFake or real images. We measure the similarity scores using the extracted feature vectors, which contain multiple features, including color, edge, and the texture of the *CSH* images. We measure both Euclidean distance scores (EDS) and cosine distance scores (CDS) to statistically compare the similarity between two extracted feature vectors and find the geometric differences between right and left *CSH* images. The EDS is defined as:

$$d\left(A, B\right) = \sqrt{\sum_{i=1}^{n} \left(A_i - B_i\right)^2} \tag{7.1}$$

Where n is the number of elements of the feature vectors, A and B are the corresponding *CSH* image vectors. d is a numerical value representing the Euclidean distance between A and B. The more similar *CSH* images, the EDS converges to 0. We also compute CDS, which is defined as:

$$\cos(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} \left(A_i\right)^2} \sqrt{\sum_{i=1}^{n} \left(B_i\right)^2}} \tag{7.2}$$

If A and B are identical, the $\cos(A, B) = 1$. Otherwise, if they are completely different $\cos(A, B) = -1$. Thus, numbers between 0 and 1 indicate a similarity score, and numbers between -1 and 0 indicate a dissimilarity score. We applied the ReLU activation function to the EDS and CDS to avoid vanishing gradient problems while training our classifiers.

The output [CDS, EDS] represents the semantic similarity between the projected representations of the two input *CSH* images.

7.2.3.3 Embedding Similarity Measures and Environmental Factors:

In addition to the reflection shape similarity (RSS) measure, we have designed similarity measures and environmental factors embedding function, which takes similarity measures [CDS, EDS], *SEF* features, and extracted (right and left) CSH features. Taking [IO, LL, LS] values from the input and annotated *SEF* values from the TDCA during training or from HEFD during testing, the similarity measures and environmental factors embedding function creates adjusted *SEF* values such as [IO', LL', LS']. Merging them with the similarity measures [CDS', EDS'] creates a row of mixed values [CDS', EDS', IO', LL', LS'] as an output. Finally, it takes vectors of (right and left) *CSH* images features and combines them in a vector for classification.

7.2.3.4 Classification:

As illustrated in Figure 20, the classification module classifies the input image, either real or DeepFake, by taking features from the embedding facility. We defined the classification network with a sequence of five blocks. The first block consists of a single BatchNormalization layer that normalizes its inputs ([CDS', EDS', IO', LL', LS']) by applying a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The following three blocks are similar. Every block consists of a sequence of a fully connected (*fc*) layer with 128 nodes, a single BatchNormalization layer followed by a ReLU activation function. The BatchNormalization layer centers the

learned features from the fully connected layer on 0, while the ReLU activation uses 0 as a pivot to keep or drop the activated channels [46]. The fifth block consists of a concatenate layer and a fully connected layer. The concatenate layer merges the fourth block's output tensor with the CSH features vector. The fully connected layer (predication layer) returns a probability distribution with two nodes and a softmax activation function for binary classification. A binary cross-entropy probabilistic loss function was used to compute the cross-entropy loss between actual and predicted labels and to measure the model's accuracy during training and testing. Eventually, it creates a binary classification result (either real or DeepFake).

7.3 Evaluations

We conducted extensive experiments using CHIEFS-DFD datasets to evaluate the performance under real-world scenarios and compare the accuracy with current state-of-the-art (SOTA) DeepFake detection methods. We demonstrate one of the environmental parameter classification results (indoor or outdoor (IO)) and evaluate *CSH* regions detection. Finally, we present the classification performances with the CHIEFS-DFD datasets using different feature extraction backbone models and various similarity measures and environmental factors.

7.3.1 Evaluation of Indoor/Outdoor Classification

The primary purpose of this experiment is to assess the CHIEFS accuracy in classifying input facial images to either indoor or outdoor environments. We combined the CHIEFS and DIODE datasets with training the indoor/outdoor classifier. Among the

20,420 images, we labeled indoor (50%) and outdoor (50%) images equally and divided 16,336 images (80%) for the training set and 4,084 images (20%) for validation and testing sets. We used MobileNetV2 inverted residuals and linear bottlenecks neural network with binary cross-entropy loss function, dense layer of two nodes, and softmax activation at the top of the network to train the indoor/outdoor classifier. All images were pre-processed and scaled between -1 and 1. We used the Glorot normal initializer from the Keras library for the default weight initialization. We trained the model on the GPU environment for 18 hours using the Google Colab Compute Engine (GCE) VM backend with (NVIDIA Tesla-P100-PCIE-16GB) model for 512 iterations with an RMSprop optimizer, batch size of 32, and learning rate of 0.001. The early stopping criterion was used with patience set to 32 to stop training when a monitored metric (validation loss) stopped improving. The indoor/outdoor classifier achieves a 94.00% success rate in predicting indoor and outdoor images. The result indicates that CHIEFS can efficiently classify input facial images into indoor or outdoor categories.

7.3.2 Evaluation of CSH Regions Detection

We evaluated the CHIEFS accuracy in detecting *CSH* regions from the facial images. We split the CHIEFS dataset (1,285 facial images containing 2,570 annotated *CSH* segmentation masks) into 1,028 images (80%) for the training set and 257 images (20%) for validation and testing sets. We used the MobileNet-V2 feature extractor model and the Single Shot Detector (SSD) to detect and return the bounding boxes of right and left *CSH* regions and class labels. We trained the *CSH* detection model on the GPU environment

Table 7: Classification performance comparison on CHIEFS-DFD dataset with different backbone models for feature extraction.

Backbone	**Accuracy**	**Loss**
CHIEFS (DenseNet-201)	96.00%	0.592
CHIEFS (Xception)	98.00%	0.242
CHIEFS (VGG-16)	98.75%	0.203
CHIEFS (ResNet-50-V2)	**99.00%**	**0.160**

for 6 hours using the Google Colab Compute Engine (GCE) VM backend with (NVIDIA Tesla-P100-PCIE-16GB) model for 1,028 iterations. We use the standard RMSprop optimizer by configuring decay and momentum to 0.9, the standard weight decay to 0.00004, an initial learning rate of 0.045, a learning rate of 0.98 per epoch, and a batch size of 32. The result demonstrates that the overall mean average precision (mAP) of detecting right and left *CSH* regions is 90.53%, the right-reflection average precision (AP) is (90.81%), and the left-reflection AP is (90.26%), both are high enough for the *CSH* detection task.

7.3.3 Classification Using Different Backbone Models for Feature Extraction

We evaluated the CHIEFS method with four different neural network backbones for feature extraction, including ResNet-50-V2, VGG-16, Xception, and DenseNet-201, using the CHIEFS-DFD dataset. After splitting the dataset with an 80:20 (training vs. validation) ratio. We trained the models on the GPU environment using the Google Colab Compute Engine (GCE) VM backend with (NVIDIA Tesla-P100-PCIE-16GB) model for 1,024 iterations with RMSprop optimizer, batch size of 8, and a learning rate of 1e-5. The early stopping criterion was used with patience set to 64 epochs to stop training when a monitored metric (validation loss) stopped improving. The results in Table 14 show the

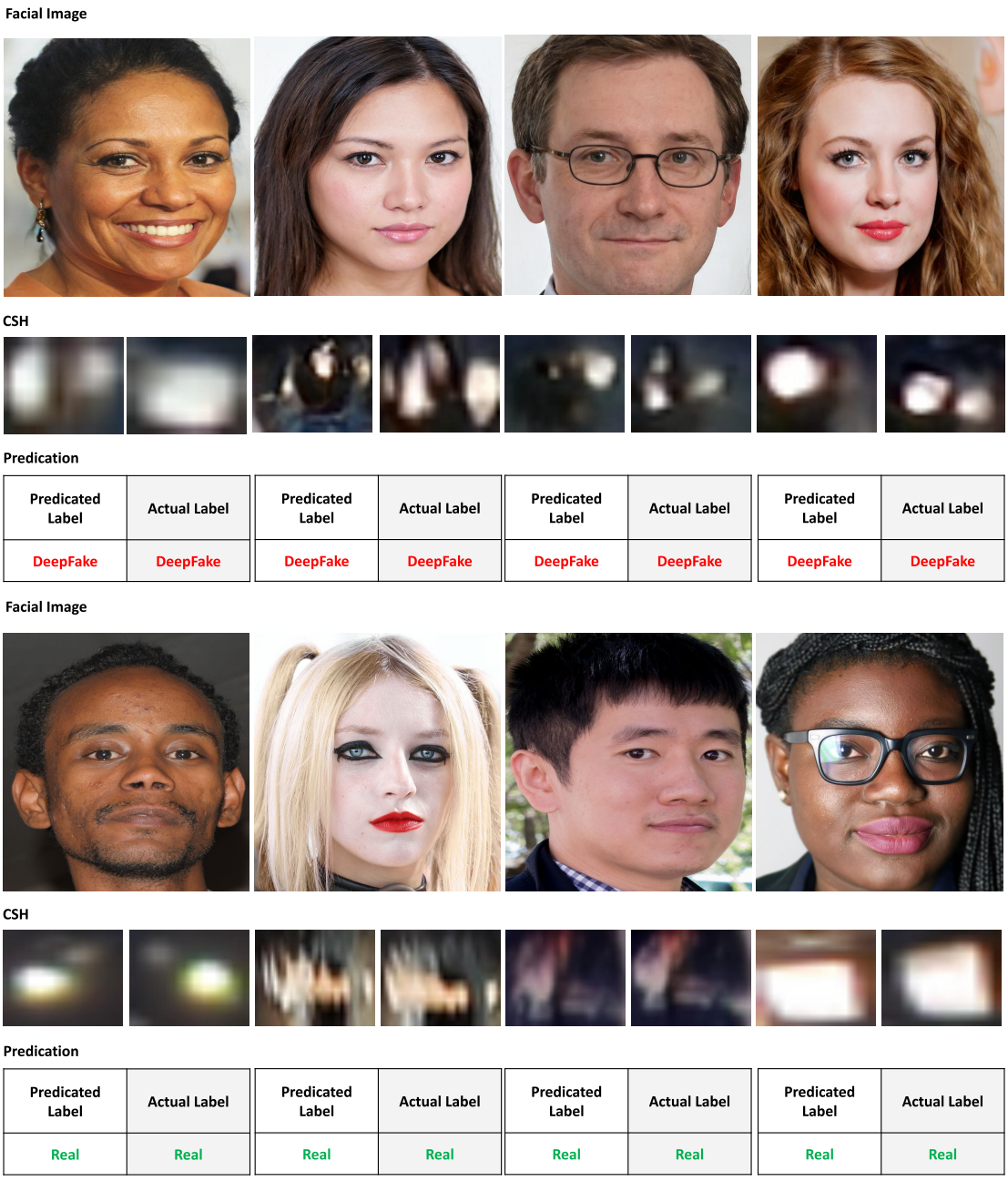


Figure 22: Sample of the CHIEFS-DFD testing dataset classification result.

Table 8: Classification performance comparison with CHIEFS-DFD dataset using different feature classifiers (i.e., CSH, CDS', EDS', IO', LL', LS') for CHIEFS (ResNet-50-V2).

Feature Classifiers	**Accuracy**
[CDS']	89.92%
[CDS', IO', LL', LS']	94.00%
[EDS']	86.05%
[EDS', IO', LL', LS']	96.00%
[CDS', EDS']	91.47%
[CSH]	93.00%
[CSH, IO', LL', LS']	97.00%
[CSH, CDS', EDS', IO', LL', LS']	**99.00%**

classification accuracy and loss of the CHIEFS method with different backbone models for feature extraction on the CHIEFS-DFD testing datasets. Overall, CHIEFS performs well with different feature extractors. For example, CHIEFS (ResNet-50-V2) is the best in both accuracy (99.00%) and loss (0.160). CHIEFS (VGG-16) is the second-best in both accuracy (98.75%) and loss (0.203). CHIEFS (Xception) is the third-best with accuracy (98.00%) and loss (0.242). Finally, CHIEFS (DenseNet-201)'s accuracy is the least (96.00%), and its loss is the highest (0.592). Figure 31 presents samples of the CHIEFS-DFD testing dataset classification results. CHIEFS detects DeepFake images with various face pose orientations, age, ethnicity, and appearances, such as makeup and accessories. Results indicate that CHIEFS performs well on realistic human visual DeepFake images.

7.3.4 Classification Using Different Feature Classifiers

Using the CHIEFS-DFD dataset, we assess different feature classifiers for CHIEFS (ResNet-50-V2). Table 15 shows that using all features, including right and left

CSH, *RSS* ([CDS', EDS']), and *SEF* ([IO', LL', LS']) for classification achieves the best performance for CHIEFS (ResNet-50-V2) (99.00%) in accuracy. However, using a single *RSS* feature alone, such as [CDS'] or [EDS'], results in low accuracy (around 89.92%) with [CDS'] and (86.05%) with [EDS']. It also demonstrates that using right and left *CSH* features achieves high accuracy (93.00%) compared with other single components such as [CDS'] and [EDS']. When *SEF* features are used with the *CSH* features, the accuracy improves to (97.00%). Similarly, when *SEF* features are used with [CDS'] and [EDS'], the accuracy also improves to (94.00%) and (96.00%), respectively. The results indicate that using a single feature alone is not a good idea, and combining various features can improve performance greatly. In addition, the *SEF* features significantly impact accuracy improvement.

7.4 Summary

We proposed a novel ML-based DeepFake detection technology named CHIEFS (Corneal-Specular Highlights Imaging for Enhancing Fake-Face Spotter). We focus on the most reflective area of a human face, eyes, using *CSH* images. We verified the hypothesis that DeepFake technologies struggle to fake reflective components in their counterfeits by using various classifiers with environmental factors embedding. We designed and implemented feature extractors, classifiers, and embedding functions using advanced DNN architectures and tested them with different GAN-generated DeepFake datasets. The experimental results show that CHIEFS achieved high accuracy 99.00% in detecting sophisticated GAN-generated DeepFake images. Note that the modular design of

CHIEFS renders itself as a complementary DeepFake detection module for any existing tools.

CHAPTER 8

MOBIDEEP: MOBILE DEEPFAKE DETECTION THROUGH MACHINE LEARNING-BASED CORNEAL-SPECULAR BACKSCATTERING

8.1 Background

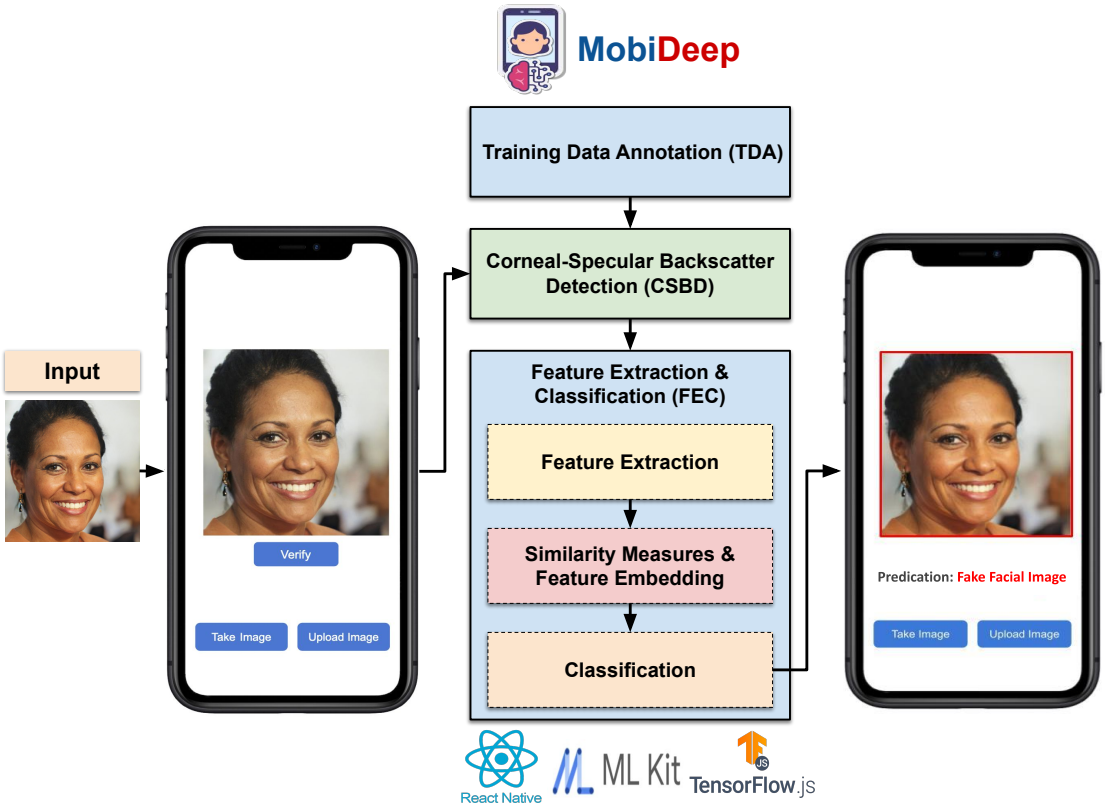


Figure 23: MobiDeep DeepFake detection method.

DeepFake techniques generating AI-leveraged fictitious human facial images, have

garnered increasing attention for their diverse usage scenarios. Although many benign applications such as funny jokes and visual humans, **DeepFake** can be malignantly used by spying on people with fake identities over social media, creating humiliating and nonconsensual fake images, spreading fake news, and planning scams and financial fraud, which becomes **serious security and privacy threat in social networks**. As DeepFake generation technologies have been improving sophisticatedly, it is getting difficult to differentiate the falsified images by bare human eyes. Furthermore, due to the recent advancement of the mobile DeepFake applications such as Reface [20], Avatarify [2], and Wombo [28], making realistic DeepFake images and videos has become astonishingly easy. Tens of millions of clips are generated every day on social networks. DeepFakes are ready to disrupt and diminish authenticity, privacy, and security for our society and worsen when the Internet becomes an immersive metaverse. The current DeepFake detection methods [68, 104, 161, 181] lack the transferability to unseen cases and become overfitted to low-quality datasets due to the limited training on low-quality videos with easy-to-detect artifacts such as shapes or visible boundaries of the fakes. Similarly, eye-based DeepFake detection methods [79, 102, 115] cannot generalize well when confronting sophisticated DeepFake media because they only consider single artifacts of eyes, either iris color, blinks, or similarity of corneal reflections on both eyes. KaiCatch [13] is the most recent DeepFake detection mobile application. However, it is a cloud-based service that takes a few days to get a classification result. Hence, real-time DeepFake detection and limitation technologies are essential to prevent the imaginable chaos that manipulates the incapacity to discern DeepFake images and videos.

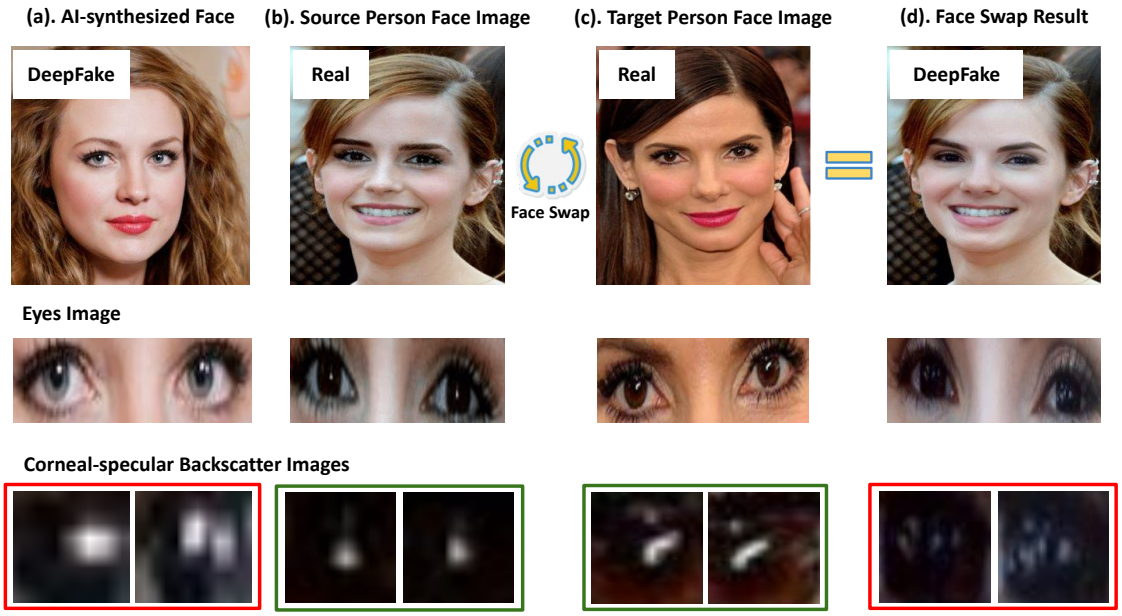


Figure 24: Samples of real and DeepFake facial images with their reflective elements (the corneal-specular backscatter images of eyes): (a) is AI-synthesized face from [88], (b) and (c) are both real, (d) is a DeepFake face generated using the Face Swapper online tool [6], Face Swapper replaces the target person's (c) facial landmarks with that of a source person (b), in the same time it preserves the source person's (b) identity.

This chapter presents a real-time, cloudless, lightweight mobile DeepFake detection technology named **MobiDeep** (**Mobi**le **Deep**Fake Detection through Machine Learning-based Corneal-Specular Backscattering), shown in Figure 23 [123]. We hypothesized that the existing DeepFake generation techniques, including replacement, editing, and synthesis, are hard to coordinate their counterfeits with the reflective elements, as presented in Figure 24 (a) and (d). Therefore, we focus on the most reflective area of a

human face, corneal-specular backscatter images of eyes. We seek the similarity and consistency of corneal-specular backscatters with multiple surrounding semantics, such as illumination and environmental conditions that are hard to fake. Thus, we extract numerous features, including corneal-specular backscatter images' color components, shapes, and textures, instead of checking a single aspect of the eyes, such as the similarity of corneal reflections on both eyes. Furthermore, we extract Facial Image Environmental Parameters (FIEP) to check the ensemble of the reflectance with the surrounding environmental factors such as indoor/outdoor, bright/dark, backgrounds, and strength and direction of light. MobiDeep embeds the *FIEP* into the feature extraction and classification process to detect the symmetricity and consistency in both eyes' color components and reflection patterns. As illustrated in Figure 23, we have implemented a cross-platform mobile application to evaluate the performance using various input parameters and lightweight Deep Neural Network (DNN) architectures. MobiDeep consists of a couple of ML components, including Training Data Annotation (TDA), Corneal-Specular Backscatter Detection (CSBD), and Feature Extraction and Classification (FEC). CSBD detects a face area and surrounding scenes from the input images to identify *CSB* images and extracts *FIEP* features. FEC extracts corneal highlight features from the *CSB* images, measures the *CSB*s symmetry and color consistency, embeds additional *FIEP* features, and classifies the *CSB* images as fake or real. We use Siamese Convolutional Neural Networks (SCNN) with three most lightweight CNN backbones including MobileNet-V2 [146], EfficientNet-B0 [164], and DenseNet-121 [80] for the feature extraction. We also create a new MobiDeep DeepFake Detection (MobiDeep-DFD) dataset, including real and fake images to annotate it with

various *CSB* information for corneal highlights segmentation. The experimental results using MobileNet-V2 with the MobiDeep-DFD dataset show that MobiDeep achieved a high accuracy (98.70%) and fast classification speed (less than 200 ms) in detecting sophisticated DeepFake images on various mobile devices.

The main contributions of this work include the following:

- A lightweight real-time mobile application is designed to cope with the cloudless ML approach by modularizing feature extraction and embedding.
- High accuracy and fast classification speed DeepFake detection application is implemented in the mobile environment.
- A high-quality DeepFake Detection dataset is collected and annotated for corneal highlight segmentation and DeepFake detection applications.
- ML methods are proposed to build an ensemble with multiple surrounding reflective features, and the impact of environmental factors on reflectance is evaluated.

8.2 Proposed Architecture

The principal objective of MobiDeep is to detect DeepFake by analyzing *CSB* images with multiple surrounding environmental parameters. MobiDeep mainly consists of Training Data Annotation (TDA), Corneal-Specular Backscatter Detection (CSBD), and Feature Extraction and Classification (FEC) modules as illustrated in Figure 25.

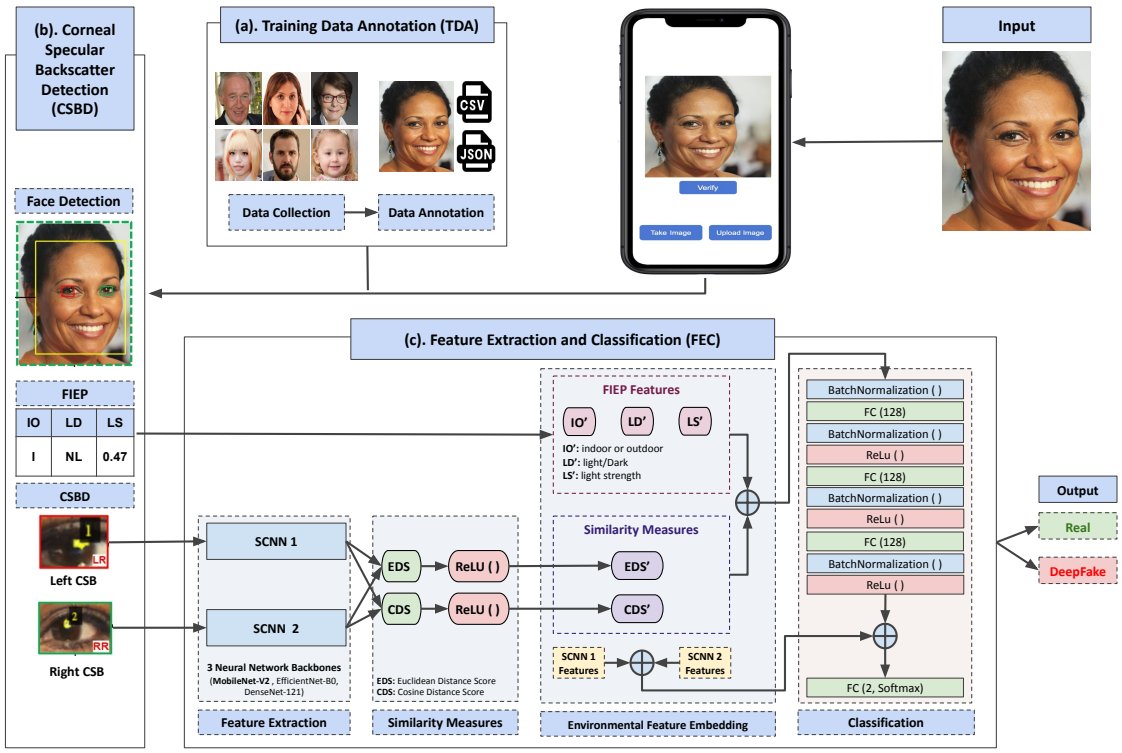


Figure 25: The block-diagram of MobiDeep DeepFake detection method.

8.2.1 Training Data Annotation (TDA)

The TDA module in Figure 25 (a) creates MobiDeep DeepFake Detection (MobiDeep-DFD) dataset by collecting and annotating real and fake facial images. The MobiDeep-DFD dataset contains the 4272 annotated corneal-specular reflection segmentation masks for 2136 facial images (two eyes per facial image). 716 real facial images were collected from different datasets, including 565 images from Flickr Faces HQ (FFHQ) dataset [87],

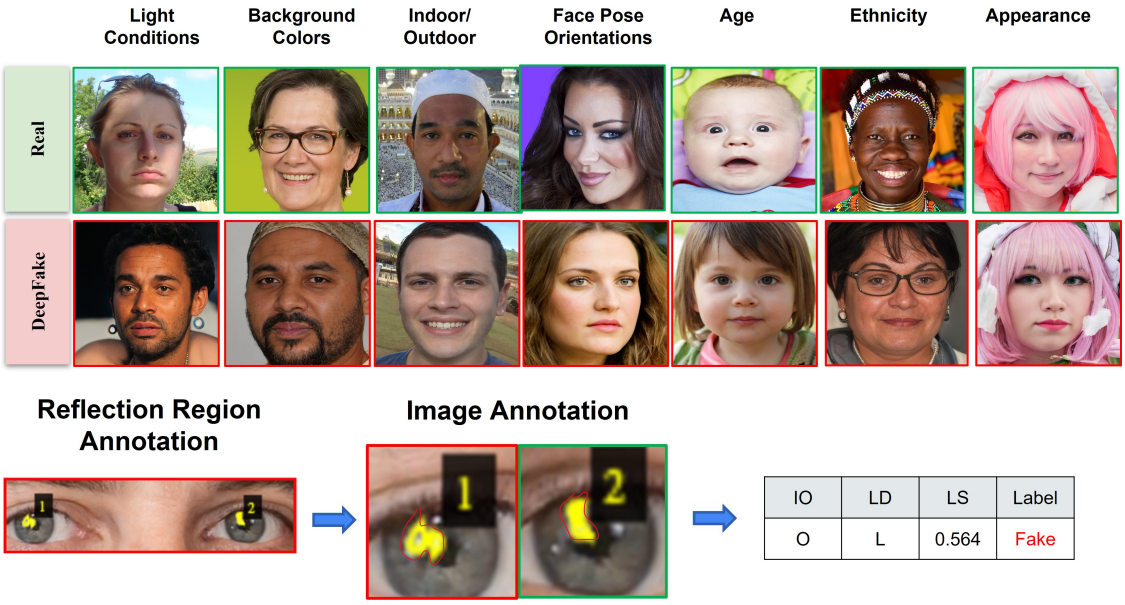


Figure 26: Image classification and annotation.

69 images from Celeb-DF dataset, 53 images from FaceForensics++ dataset, and 29 images from DFDC dataset. Similarly, 1420 fake facial images were acquired from various DeepFake detection datasets using various DeepFake generation tools, including 569 face synthesis DeepFake images from StyleGAN2 [88], 431 images from the Celeb-DF dataset, 369 images from the FaceForensics++ datasets, and 51 images from DFDC dataset. As presented in Figure 26, the MobiDeep-DFD dataset contains fake and real facial images in high and low quality with various facial image environmental parameters (FIEP), including illumination conditions, background colors, indoor or outdoor settings, face poses orientations, age, ethnicity, and appearances (e.g., wearing makeup and accessories). As illustrated in Figure 26, the MobiDeep-DFD dataset has two types

of annotation for each facial image, including the *Reflection Region Annotation* to define the shapes and locations of *CSB* regions, classifying them into right-reflection and left-reflection classes and the *Image Annotation* to identify the image label (either real or fake), along with *FIEP*, including indoor or outdoor (IO), light or dark (LD), and light strength (LS).

8.2.2 Corneal Specular Backscatter Detection (CSBD)

The CSBD module in Figure 25 (b) performs face detection, FIEP feature extraction, and *CSB* localization. CSBD uses a pre-trained MediaPipe Face Detection model to locate a human face in an image and provide its associated position, size, and orientation. CSBD is also responsible for detecting *FIEP*, including indoor or outdoor (IO), light or dark (LD), and light strength (LS). We train a MobileNet-V2 model on the dense indoor and outdoor depth (DIODE) [169] dataset and labeled facial images from the MobiDeep-DFD dataset to classify input images into indoor or outdoor. Our indoor/outdoor dataset includes 20420 images by merging the DIODE and MobiDeep-DFD datasets. To calculate the light strength (LS) of the input facial image, first, we convert the input image color space to LAB format. The L channel is independent of color information in the LAB color space and only encodes lightness (intensity). The other two channels A and B encode color. Next, we extract the L channel and normalize it by dividing all pixel values by the maximum pixel value. Finally, it returns the input image's mean of light strength (LS). Analyzing the distribution of the obtained light strength values from our dataset and using the standard deviation, we have a standard way of knowing what image has normal

light intensity and which has high light or dark. The input image will be classified as normal light if its mean light strength (LS) is in the range of 0.419 to 0.637, high light if it is more than 0.637, or dark if it is less than 0.419. CSBD also detects right and left *CSB* from the detected face and generates high-quality *CSB* images. We train the CSBD model using the MobileNet-V2 and its modified Single Shot Detector (SSD) version, known as SSDLite, to detect and return the bounding boxes of right and left *CSB* regions and class labels.

8.2.3 Feature Extraction and Classification (FEC)

Using the right and left *CSB* images extracted from the CSBD module, ***the FEC module in Figure 25 (c)*** performs feature extraction, measures similarity scores, embeds the similarity score with FIEP, and does classification. FEC can extract features from each *CSB* image by using a Siamese Convolutional Neural Network (SCNN) model with various CNN backbones.

8.2.3.1 Feature Extraction

Table 9: Size and parameters of feature extraction backbone models.

Backbones	Size (MB)	Parameters
MobileNet-V2 [146]	**14**	**3,538,984**
EfficientNet-B0 [164]	**29**	**5,330,571**
DenseNet-121 [80]	**33**	**8,062,504**
ResNet-152 [73]	232	60,419,944
VGG-16 [155]	528	138,357,544

To obtain features from the *CSB* images, the feature extraction model runs a couple of SCNN models in parallel for left and right *CSB*s. The proposed SCNN model consists of two identical CNNs with the same weights to extract deep learning features from the *CSB* inputs. It takes various CNN backbones, including MobileNet-V2, EfficientNet-B0, DenseNet-121, ResNet-152, and VGG-16. As shown in Table 9, we have picked the three most lightweight neural network architectures in both the package size and the number of parameters to cope with the resource constraints (GPU, CPU, memory, and communication) on the mobile devices. Each SCNN module accepts an RGB image of size 224×224 pixels from the CSBD module. Two SCNNs are both used feedforwards to extract features using a global max-pooling layer by removing the fully-connected layer at the top of every network ($include_top$= False). We do not need activation and classes because we only use the backbone models for feature extraction and compare their output at the end by measuring similarity scores.

8.2.3.2 Similarity Measures

As illustrated in Figure 26, *CSB*s are detected in various shapes. According to illumination conditions and background settings, the *CSB*s can be deformed in different colors and blended into the background. For example, in Figure 26, *CSB* shapes of the left and right eyes are different even if the person is looking in the same direction. Also, *CSB*s can be occluded by glasses, eyelids, or eyelashes, and only a tiny portion of reflection can be visible. Hence, the similarity measures of a single factor such as the *CSB* shape or color on both eyes alone cannot be a strong indicator for classifying fake or real images.

We measure the similarity scores using the extracted feature vectors, which contain multiple features, including color, edge, and the texture of the *CSB* images. We measure both Euclidean distance scores (EDS) and cosine distance scores (CDS) to statistically compare the similarity between two extracted feature vectors and find the geometric differences between right and left *CSB* images. We applied the ReLU activation function to the EDS and CDS to avoid vanishing gradient problems while training our classifiers. The output [CDS, EDS] generated by SCNN execution represents the semantic similarity between the projected representations of the two input *CSB* images. In addition to the similarity measures, we have designed a feature embedding facility to enhance the feature classification result by applying the environmental factors (FIEP). It is a configurable platform to add multiple embedding functions. For MobiDeep, we have implemented an Environmental Feature Embedding (EFE) function, which takes a few FIEPs, including indoor/outdoor (IO), light/dark (LD), and light strength (LS). As shown in Figure 26, for simplicity, we take boolean values for IO and LD and numerical values from 0 to 1 for LS. These FIEPs can be added and merged according to the requirements. Taking a row of [IO, LD, LS] from the input and annotated FIEP values from the TDA, EFE create adjusted FIEP values such as [IO', LD', LS']. Merging them with the similarity measures [CDS', EDS'] creates a row of 5 col numerical values [CDS', EDS', IO', LD', LS'] as an output. EFE function also takes the right and left CSB features vectors and combines them in one vector for classification.

8.2.3.3 Classification

As illustrated in Figure 25, the classification module finally classifies images to either real or fake by taking a row of 5 column values [CDS', EDS', IO', LD', LS'] created from the EFE function. We defined the classification network with a sequence of five blocks. The first block consists of a single BatchNormalization layer that normalizes its inputs by applying a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The following three blocks are similar. Every block consists of a sequence of a fully connected (*fc*) layer with 128 nodes, a single BatchNormalization layer followed by a ReLU activation function. The BatchNormalization layer centers the learned features from the fully connected layer on 0, while the ReLU activation uses 0 as a pivot to keep or drop the activated channels [46]. The fifth block consists of two layers, a concatenate layer to merge the fourth block's output tensor with the right and left CSB features tensor, and a fully connected layer (predication layer) with two nodes and a softmax activation function to return a probability distribution for binary classification. A binary cross-entropy probabilistic loss function is used to compute the cross-entropy loss between actual labels and predicted labels and measure how accurate the model is during training and testing. Eventually, it creates a binary classification result (either real or fake) as an output.

8.3 Evaluations

We conducted extensive experiments on the MobiDeep implementation in Android, iOS, and web applications to evaluate the performance under real-world scenarios

and compare the accuracy and speed with current state-of-the-art (SOTA) DeepFake detection methods.

8.3.1 Evaluations of Execution Speed

The primary goal of the experiments is to assess the feasibility of MobiDeep usage on mobile devices by checking the performance of MobiDeep classification speed on both GPU and CPU environments and find suitable feature extractor models among MobileNet-V2, EfficientNet-B0, and DenseNet-121 for the mobile application.

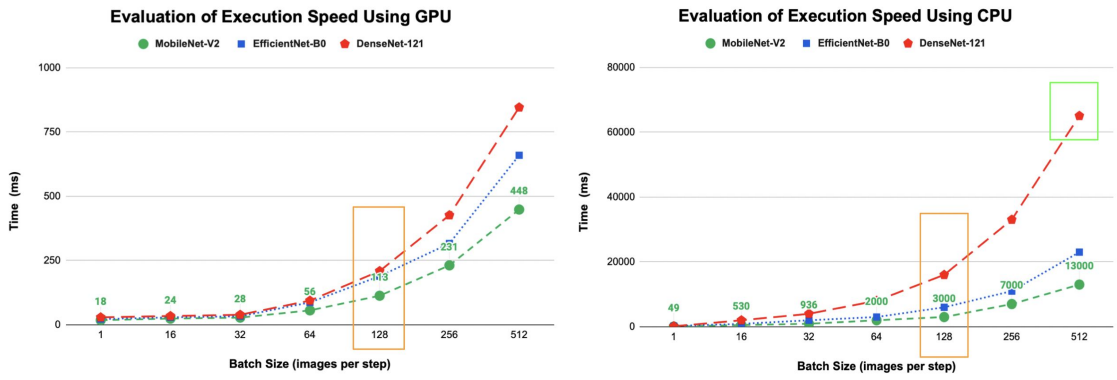


Figure 27: Evaluation of testing speed with GPU and CPU using different backbone models for feature extraction.

Figure 27 shows the testing speed on the GPU environment using the Google Colab Compute Engine (GCE) VM backend (with NVIDIA Tesla-P100-PCIE-16GB) and 8-cores CPU. Testing batch sizes (i.e., images per step) increases, and the delay for all models on both GPU and CPU increases. MobileNet-V2 is the fastest, and DenseNet-121 is the slowest. As presented in the left-hand panel of Figure 27 for the typical batch size of 128 images with GPU, all models can evaluate within 250 ms. In contrast, the right-hand

panel of Figure 27 shows that with 8-core CPU, MobileNet-V2 and EfficientNet-B0 both models can classify a batch size of 128 images in 3 seconds and 6 seconds, respectively. The DenseNet-121 delay grows faster than other models on an 8-core CPU. MobileNet-V2 offers the fastest evaluation speed.

Table 10: Detection speed on Android and iOS.

Backbones	**Type**	**FEC Delay (ms)**	**Total Delay (ms)**
MobiDeep (DenseNet-121)	Galaxy S9	69.68	191.31
	iPhone 11	72.83	197.36
MobiDeep (MobileNet-V2)	Galaxy S9	45.82	167.45
	iPhone 11	47.59	172.12

As shown in Table 10, we assessed the feasibility of MobiDeep on mobile devices, including Android and iOS. We have built a real-time, cloudless, lightweight cross-platform mobile application using React Native user interface software framework and TensorFlow.js hardware-accelerated JavaScript library for deploying our ML models on mobile devices. Samsung Galaxy S9 (SM-G960F) comes with a 2.7 GHz Octa-Core processor, 64 GB memory, 4 GB RAM, and a 3000 mAh battery. iPhone 11 has an A13 Bionic chip (with 6-core CPU, 4-core GPU, and 8-core Neural Engine), 128 GB memory, 4 GB RAM, and a built-in rechargeable lithium-ion battery. We collected the average execution speed of CSBD and FEC with two deep neural network feature extraction architectures (MobileNet-V2 and DenseNet-121). MobiDeep (MobileNet-V2) has a low average execution delay on both Samsung Galaxy S9 (167.45 ms) and iPhone 11 (172.12 ms) compared to MobiDeep (DenseNet-121)'s average execution speed was (191.31 ms) on Samsung Galaxy S9 and (197.36 ms) on iPhone 11. MobiDeep operates efficiently

within 200 ms on Android and iOS mobile devices with an easy-to-use, stand-alone, and lightweight mobile application.

8.3.2 Classification Using Different Backbone Models for Feature Extraction

Table 11: Classification accuracy.

Backbones	**Accuracy**	**Loss**
MobiDeep (EfficientNet-B0)	91.27	0.185
MobiDeep (DenseNet-121)	97.35	0.053
MobiDeep (MobileNet-V2)	**98.70**	**0.029**

The primary goal of the experiments is to assess the feasibility of MobiDeep usage on mobile devices by checking the performance of MobiDeep classification accuracy with various feature extractor models. As shown in Table 14, we used the three most lightweight CNN backbones (MobileNet-V2, EfficientNet-B0, and DenseNet-121) for feature extraction. Three classifiers were trained on the MobiDeep-DFD training dataset and tested on the MobiDeep-DFD testing dataset. Table 14 shows that classifier accuracy with different feature extractors. MobiDeep is highly effective (over 90%) in detecting DeepFake images. MobiDeep (MobileNet-V2) is the best in both accuracy (98.70%) and loss (0.029). MobiDeep (DenseNet-121) is the second-best in both accuracy (97.35%) and loss (0.053). Hence, MobileNet-V2 and DenseNet-121 can be used for feature extraction without any significant difference. However, MobiDeep (EfficientNet-B0)'s accuracy is the least (91.27%), and MobiDeep (EfficientNet-B0)'s loss is the highest (0.185). Hence, EfficientNet-B0 may not be recommended for MobiDeep's feature extraction.

8.4 Summary

This chapter presented the design and development of a real-time, cloudless, lightweight mobile DeepFake detection technology named **MobiDeep** (**Mobi**le **Deep**Fake Detection through Machine Learning-based Corneal-Specular Backscattering). Focusing on the hypothesis that the existing DeepFake methods, including replacement, editing, and synthesis, are hard to coordinate their counterfeits with the reflective elements, MobiDeep took a novel approach using the corneal-specular backscatter images of human eyes. It evaluates the similarity and consistency with multiple surrounding environment features, Facial Image Environmental Parameters (FIEP), including color components, shapes, and textures, instead of merely checking the similarity between eye reflection shapes. We have implemented a cross-platform mobile application to evaluate the performance using various input parameters and lightweight Deep Neural Network (DNN) architectures. The experimental results show that MobiDeep achieved the high accuracy (98.70%) and rapid detection speed (less than 200 ms) in detecting sophisticated DeepFake images using MobileNet-V2 with the MobiDeep-DFD dataset.

CHAPTER 9

READFAKE: REFLECTION AND ENVIRONMENT-AWARE DEEPFAKE DETECTION

9.1 Background

The AI-fueled production and manipulation techniques of fictitious human facial images, DeepFake [118], have accomplished notable advancement. Due to the sophisticated DeepFake generation technologies [88,95,136], it is getting harder to distinguish the forged images. Despite many benign applications such as fun memes, visual effects, and realistic avatars, the generated fake media can be malignantly used by spreading misinformation on social media, creating deception for identity theft, and causing manipulation of election security. It may become a critical risk to authenticity, privacy, and security for our society [130]. Hence, appropriate DeepFake detection technologies are essential to mitigate potential malignant risks.

Recently, DNN-based DeepFake detection methods have been developed by extracting discriminative features from facial images. For example, [166] used Convolutional Neural Network (CNN) to detect synthesis faces. [77] proposed a pairwise learning model to distinguish the features between fake and authentic images. [181] created multi-attentional network architecture to capture local discriminative features from multiple face regions. Another technique by [161] used facial geometric information and its

temporal characteristics to classify DeepFake videos. [48] used the photoplethysmography optical measurement technique to monitor and examine the subtle biological signals hidden in portrait videos and run the collected signatures through a classifier to determine whether the video is real or fake. [51] examined the spatial, temporal, and spectral consistency of eyes and gazes in five domains (e.g., visual domain, geometric domain, temporal domain, etc.) to classify DeepFake videos. However, the existing methods suffer from overfitting and lack of detection generalization on unseen cases because they were trained on datasets containing few low-quality videos generated with a single model and fewer subjects. Likewise, eye-based DeepFake detection methods [68, 79, 102, 115] cannot generalize well when confronting sophisticated DeepFake media because they only consider single artifacts of eyes, either iris color, blinks, or similarity of corneal specular reflections on both eyes.

We propose a novel **R**eflection and **E**nvironment-**A**ware **D**eep**Fake (READFake)** detection technology. We focus on the similarity and consistency of specular highlights on various body parts such as eyes, nose, cheeks, etc., as shown in Figure 28, along with multiple surrounding semantics. For example, illumination and environmental conditions are hard to forge, upon the hypothesis that the existing DeepFake creation methods, including reenactment, replacement, and synthesis, are hard to coordinate their counterfeits with the reflective components along with the given environmental mapping. Originally, the project focused on Corneal Specular Highlights (*CSH*) on eyes, the most reflective areas of human faces. However, as shown in Figure 28, it may not always be available due to the limitations of the images with blurriness, profile faces, low-quality images, and occlusions

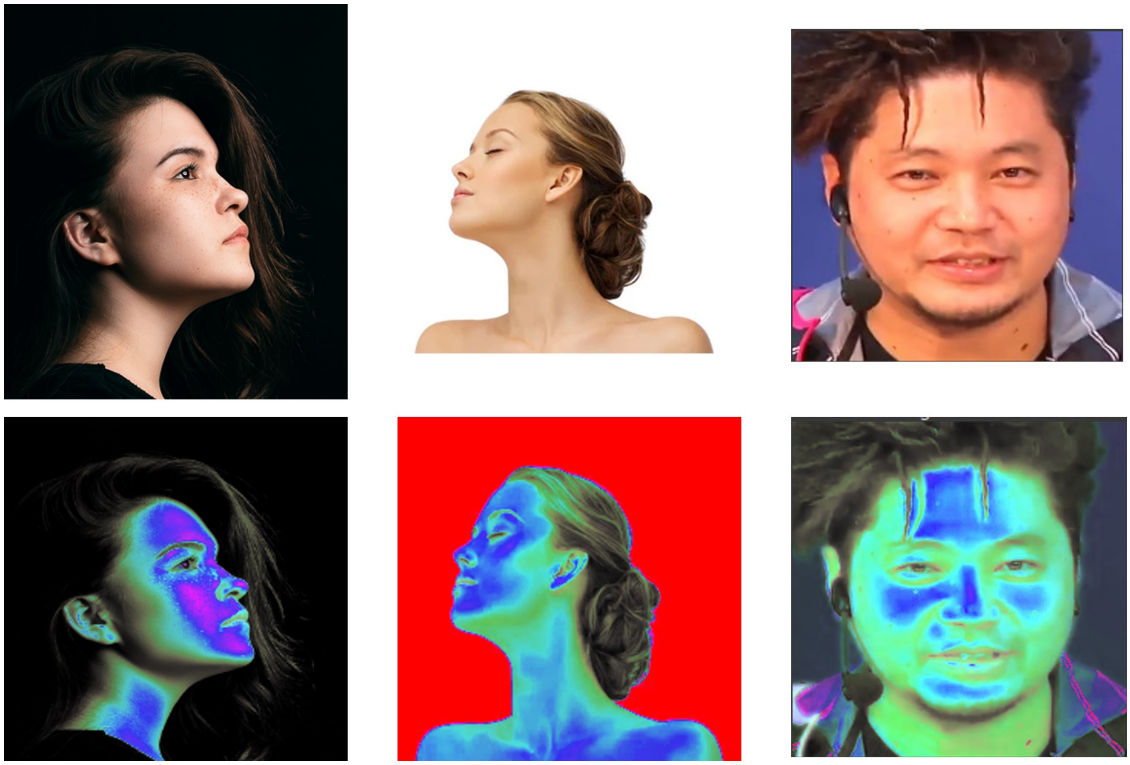


Figure 28: Body reflection highlights from various facial images (profile, front, open or closed eyes).

(hair occludes one eye while leaving the other visible). We collect diverse features such as color components, shapes, and textures from specular highlight images from various facial and body parts to check the coordination with the surrounding environmental factors, including indoor/outdoor, bright/dark backgrounds, and light strength. As illustrated in Figure 29, READFake consists of Machine Learning (ML) components, including Training Data Annotation (TDA), Reflections and Environmental Factors Detection (REFD), and Feature Extraction, Embedding, and Classification (FEEC). The TDA creates and annotates a new DeepFake detection dataset named READFake. The READFake dataset

includes real and DeepFake facial images annotated with various specular highlights and environmental information. The REFD detects facial image environmental parameters *FIEP* (e.g., indoor/outdoor, light level, and light strength) and identifies specular highlights from various body parts. The FEEC leverages various configurable neural network backbones, including ResNet152-V2 [74], DenseNet-169 [80], EfficientNet-B1 [164], and Inception-V3 [162] to extracts features from specular highlight images. It checks the right and left eyes' *CSH* and *BSH* symmetry and consistency with embeds reflections and environmental features. We have conducted experiments with the existing DeepFake detection datasets, including FaceForensics++ [142], Celeb-DF [103], and DFDC [56] in addition to the READFake dataset to validate the accuracy of READFake. The results show that the accuracy (99.0%) READFake achieved is better than state-of-the-art (SOTA) methods. Furthermore, the modular design of READFake makes it available as a complementary DeepFake detection module for the existing tools.

The main contributions of this work include:

- A new DeepFake detection dataset is created by collecting and annotating diverse facial images.
- A ML method is proposed to build an ensemble with various reflection features from the diverse body and facial parts instead of a single feature from the human eyes.
- The impact of environmental factors on reflectance is studied by collecting various environmental information.

- An investigation study on various SOTA DeepFake datasets and feature extractors is performed.

- A modular design is created for feature extraction and embedding for porting to other existing tools as a complementary module.

9.2 Proposed Architecture

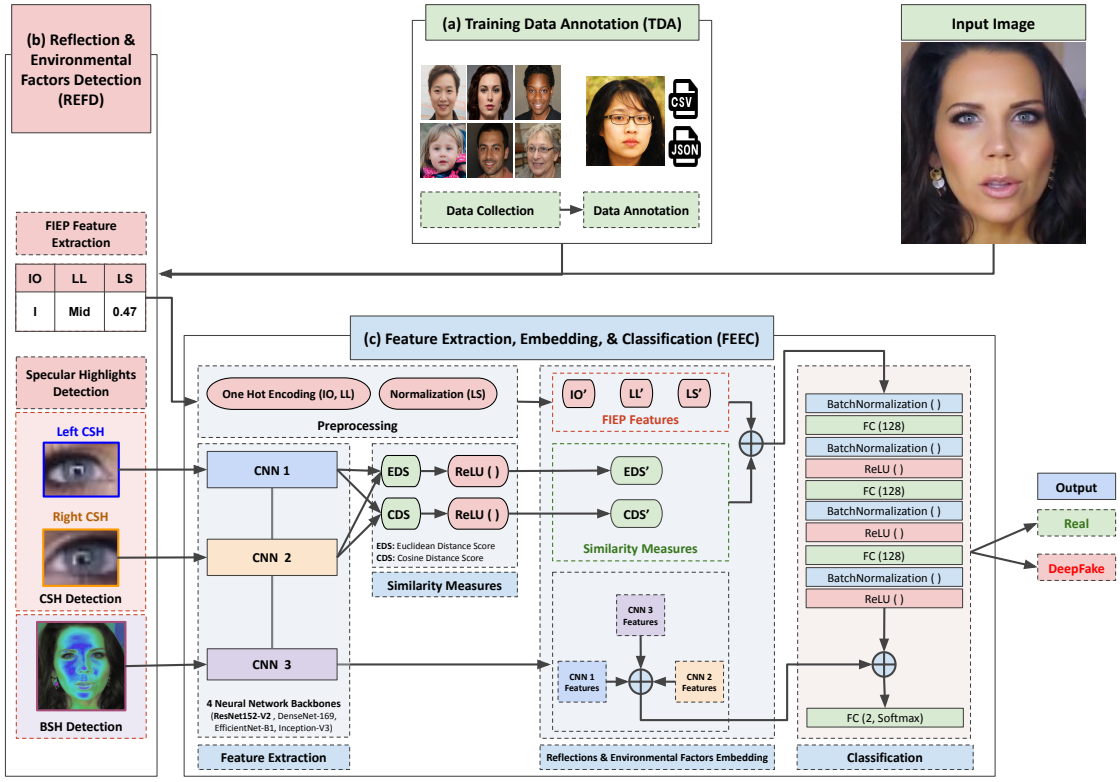


Figure 29: The READFake architecture block-diagram.

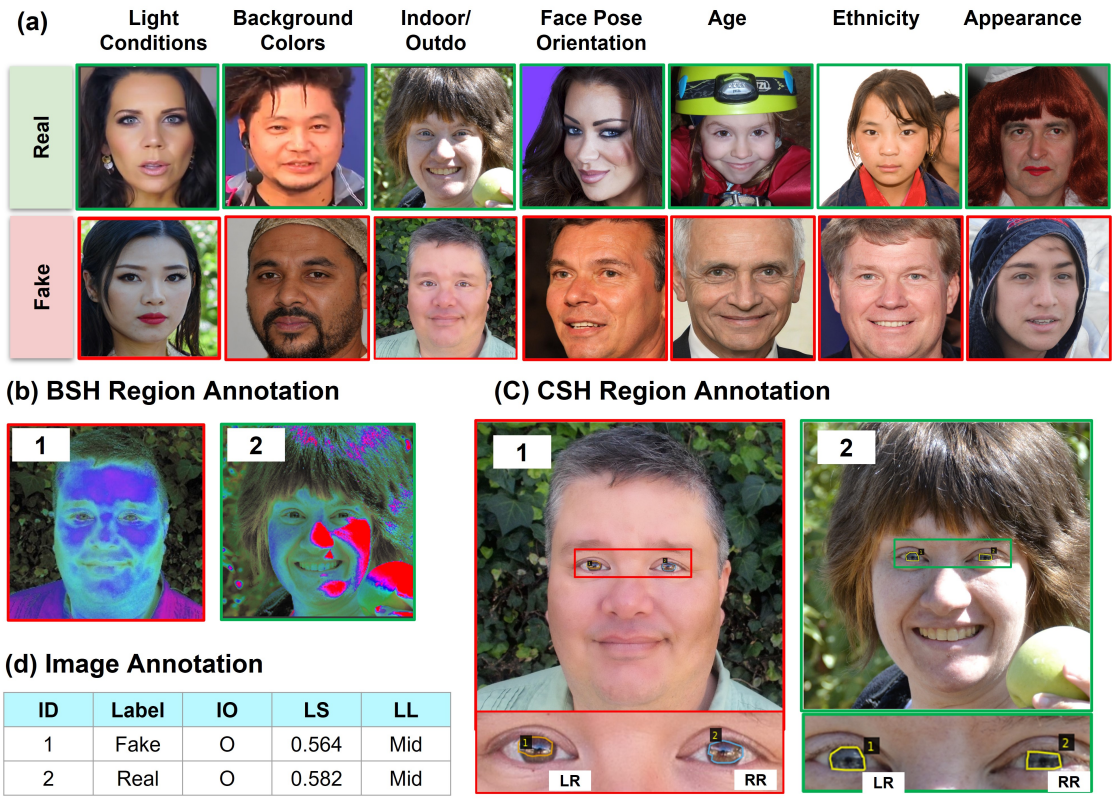
READFake is a reflection and environment-aware DeepFake detection technology that aims to analyze specular highlights on various body parts and check the coordination

with the surrounding environmental parameters. We design READFake in a hierarchical structure and organize its components into three distinct modules. READFake mainly consists of Training Data Annotation (TDA), Reflections and Environmental Factors Detection (REFD), and Feature Extraction, Embedding, and Classification (FEEC) modules as illustrated in Figure 29. The proposed structure supports agile updates of each component according to specific use cases and makes READFake available as a complementary DeepFake detection module for other existing tools.

9.2.1 Training Data Annotation (TDA)

The current DeepFake detection datasets (e.g., UADFV, FaceForensics++, Celeb-DF, DFFD, and DFDC) do not contain specular highlights annotation or facial image environmental parameters information. Thus, the main responsibility of ***the TDA module in Figure 29 (a)*** is to create READFake dataset [19] by collecting and annotating real and DeepFake facial images. We manually label the most reflective regions on the human body, CSH, and provide facial image-specific environmental information using the VGG Image Annotator (VIA) software [61]. The READFake dataset contains 2136 facial images in high and low resolution. 716 real facial images were collected from different datasets, including 565 images from Flickr Faces HQ (FFHQ) dataset [87], 69 images from Celeb-DF dataset, 53 images from FaceForensics++ dataset, and 29 images from DFDC dataset. We acquired 1420 DeepFake facial images from various DeepFake detection datasets and human visual DeepFake generation tools, including 569 face synthesis DeepFake images from StyleGAN2 [88] and StyleGAN3 [86]. Also, we collected 431

images from the Celeb-DF dataset, 369 images from the FaceForensics++ dataset with Face2Face and FaceSwap methods, and 51 images from the DFDC dataset.



ID	Label	IO	LS	LL
1	Fake	O	0.564	Mid
2	Real	O	0.582	Mid

Figure 30: Environmental parameter samples and annotations in READFake dataset.

As presented in Figure 30 (a), the READFake dataset contains DeepFake and real facial images in various resolutions with different environmental parameters, including illumination conditions, background colors, indoor or outdoor settings, face pose orientations, age, ethnicity, and appearances (e.g., wearing makeup and accessories). We annotated the READFake dataset with 3 different types. The *Body Specular Highlight*

(BSH) region annotation in Figure 30 (b) identifies highlight patterns from various reflective body regions except eyes. The *CSH* region annotation in Figure 30 (c) defines the shapes and locations of *CSH* and classifies CSHs into right-reflection and left-reflection classes. The *Image Annotation* in Figure 30 (d) identifies the image label (either Real or Fake), along with *FIEP* parameters, including indoor or outdoor (IO), light level (LL), and light strength (LS). The READFake dataset contains the 4272 annotated *CSH* segmentation masks for 2136 facial images (two eyes per facial image). In addition, 1779 images (83.30%) are labeled as indoor, and 357 images (16.70%) are labeled as outdoor. Furthermore, collecting and analyzing the distribution of READFake facial images' LS values (explained in Subsection 9.2.2) results in different LL classes (1414 mid images (66.20%), 438 low images (20.50%), and 284 high images (13.30%)).

9.2.2 Reflections and Environmental Factors Detection (REFD)

The REFD module in Figure 29 (b) performs two major tasks, including *FIEP* feature extraction and specular highlights detection. The *FIEP* parameters include *IO*, *LS*, and *LL*. We train a MobileNet-V2 model on the Dense Indoor and Outdoor Depth (DIODE) dataset [169] and labeled facial images from the READFake dataset to classify the IO of an input image (totaling 20420 images). To calculate the *LS*, we convert the input image's color space into a LAB format. The *L* channel is independent of color information in the LAB color space and only encodes intensity. The other two channels *A* and *B* encode color. Then, we extract the *L* channel and normalize it by dividing all pixel values by the maximum pixel value to have an *LS* value of the input image. Using

the *LS* value, we identify an *LL* into the low, mid, and high classes (e.g., according to the *LS* distribution, the *LL* is low if *LS* is less than 0.419, high if *LS* is greater than 0.637, and a mid if it is in between). The specular highlights detection consists of BSH and CSH detections. We train the *CSH* detection model using the MobileNetV2-SSDLite [146] to detect the bounding boxes of right and left *CSH* regions and class labels. To identify the BSH detection, it remaps the input image's primary colors into HSV (hue, saturation, value) color space dimensions. *Hue* specifies the angle of the color from 0 to 360 degrees, *saturation* controls the amount of color used from 0 to 100 percent, and *value* controls the brightness of the color from 0 (black) to 100. The BSH detection highlights the brightest point with a high value and low saturation.

9.2.3 Feature Extraction, Embedding, and Classification (FEEC)

Using specular highlight features and *FIEP* extracted from the REFD module, ***the FEEC module in Figure 29 (c)*** performs four primary functions, including deep hierarchical feature extraction, similarity scores measure, reflections and environmental factors embedding, and classification.

9.2.3.1 Feature Extraction

As shown in Figure 29 (c), the CNN models with the same weights and network architecture in the **FEEC** module receive the right and left *CSH* and BSH images in parallel. Various configurable neural network backbones, including ResNet152-V2, DenseNet-169, EfficientNet-B1, and Inception-V3, are used for the feature extraction. The CNNs use feedforwards to extract features using a global max-pooling layer by removing the

fully-connected layer at the top of every network ($include_top$= False). We do not need activation and classes because we only use the backbone models for feature extraction. Then, we use the right and left *CSH* features to measure a similarity score.

9.2.3.2 Similarity Measures

CSH can be detected in various shapes, which can be deformed in different colors according to illumination conditions and blended into the background. Also, *CSH* can be occluded by glasses, eyelids, or eyelashes, and only a tiny portion of the reflection can be visible. Hence, the similarity measures of a single factor, such as the shape or color of the *CSH* alone, cannot be a strong indicator for classifying fake or real images. We measure the similarity scores using the extracted feature vectors, which contain multiple features, including color, edge, and the texture of the *CSH* images. We measure both Euclidean distance scores (EDS) and cosine distance scores (CDS) to statistically compare the similarity between two extracted feature vectors and find the geometric differences between right and left *CSH* images. We applied the ReLU activation function to the EDS and CDS to avoid vanishing gradient problems while training our classifiers. The output [CDS, EDS] represents the semantic similarity between the projected representations of the two input *CSH* images.

The EDS is defined as:

$$d\left(A, B\right) = \sqrt{\sum_{i=1}^{n} \left(A_i - B_i\right)^2} \tag{9.1}$$

Where n is the number of elements of the feature vectors, A and B are the corresponding *CSH* image vectors. d is a numerical value representing the Euclidean distance between A and B. The more similar *CSH* images, the EDS converges to 0. We also compute CDS, which is defined as:

$$\cos(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{9.2}$$

If A and B are identical, the $\cos(A, B) = 1$. Otherwise, if they are completely different $\cos(A, B) = -1$. Thus, numbers between 0 and 1 indicate a similarity score, and numbers between -1 and 0 indicate a dissimilarity score.

9.2.3.3 Feature Embedding

We implemented an environmental feature embedding function, which takes similarity measures [CDS, EDS], *FIEP* features and extracted (right and left) CSH and BSH features. Taking [IO, LL, LS] values from the input and annotated *FIEP* values from the TDA during training or from REFD during testing, it creates adjusted *FIEP* values such as [IO', LL', LS'] using one hot encoding and normalization. Merging them with the similarity measures [CDS', EDS'], creates a row of mixed values [CDS', EDS', IO', LL', LS'] as an output. Finally, it takes the obtained feature vectors of (right and left) *CSH* and *BSH* images and combines them in a vector for classification.

9.2.3.4 Classification

It classifies the real or DeepFake input image by taking features from the embedding vector created by the environmental feature embedding function. We defined the classification network with a sequence of five blocks. The first block consists of a single BatchNormalization layer that normalizes its inputs ([CDS', EDS', IO', LL', LS']) by applying a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The following three blocks are similar. Every block consists of a sequence of a fully connected (*fc*) layer with 128 nodes, a single BatchNormalization layer followed by a ReLU activation function. The BatchNormalization layer centers the learned features from the fully connected layer on 0, while the ReLU activation uses 0 as a pivot to keep or drop the activated channels. The fifth block consists of a concatenate layer and a fully connected layer. A concatenate layer merges the fourth block's output tensor with the specular highlights features vector. A fully connected layer (predication layer) returns a probability distribution with two nodes and a softmax activation function for binary classification. A binary cross-entropy probabilistic loss function is used to compute the cross-entropy loss between actual and predicted labels and measure the accuracy of the model during training and testing. Eventually, it creates a binary classification result (either Real or DeepFake) as an output.

9.3 Evaluations

We conducted extensive experiments using various DeepFake datasets to evaluate the performance under real-world scenarios and compare the accuracy with current

SOTA DeepFake detection methods. We demonstrate one of the environmental parameter classification results (indoor or outdoor (IO)) and evaluate *CSH* regions detection. Then, we present results with the READFake and FaceForensics++ datasets for training and testing. Also, we show the cross-dataset classification performance evaluation results of the READFake pre-trained model compared with the Celeb-DF and DFDC datasets, respectively. Finally, we present the classification performances using different feature extraction backbone models and different reflections and environmental factors.

9.3.1 Evaluation of Indoor/Outdoor Classification

The primary purpose of this experiment is to assess the READFake accuracy in classifying input facial images to either indoor or outdoor environments. We combined the READFake and DIODE datasets with training the indoor/outdoor classifier. Among the 20420 images, we labeled indoor (50%) and outdoor (50%) images equally and divided 16336 images (80%) for the training set and 4084 images (20%) for validation and testing sets. We used MobileNetV2 inverted residuals and linear bottlenecks neural network with binary cross-entropy loss function, dense layer of two nodes, and softmax activation at the top of the network to train the indoor/outdoor classifier. All images were pre-processed and scaled between -1 and 1. We trained the model on the GPU environment for 18 hours using the Google Colab Compute Engine (GCE) VM backend with (NVIDIA Tesla-P100-PCIE-16GB) model for 512 iterations with an RMSprop optimizer, batch size of 32, and learning rate of 1e-3. The indoor/outdoor classifier achieves a 94.00% success rate in predicting indoor and outdoor images. The result indicates that READFake can efficiently

classify input facial images into indoor or outdoor categories.

9.3.2 Evaluation of CSH Regions Detection

We evaluated the READFake accuracy in detecting *CSH* regions from the facial images. We have created a new dataset, READFake dataset [19] that includes 2136 facial images containing 4272 annotated *CSH* segmentation masks. We split it into 1708 images (80%) for the training set and 428 images (20%) for validation and testing sets. We used the MobileNet-V2 feature extractor model and the Single Shot Detector (SSD) to detect and return the bounding boxes of right and left *CSH* regions and class labels. We trained the *CSH* detection model on the GPU environment for 6 hours using the Google Colab Compute Engine (GCE) VM backend with (NVIDIA Tesla-P100-PCIE-16GB) model for 1028 iterations. We use the standard RMSprop optimizer by configuring decay and momentum to 0.9, the standard weight decay to 0.00004, an initial learning rate of 0.045, a learning rate of 0.98 per epoch, and a batch size of 32. The result demonstrates that the overall mean average precision (mAP) of detecting right and left *CSH* regions is 90.53%, the right-reflection average precision (AP) is (90.81%), and the left-reflection AP is (90.26%), both are high enough for the *CSH* detection task.

9.3.3 Evaluation on READFake and FaceForensics++ Datasets

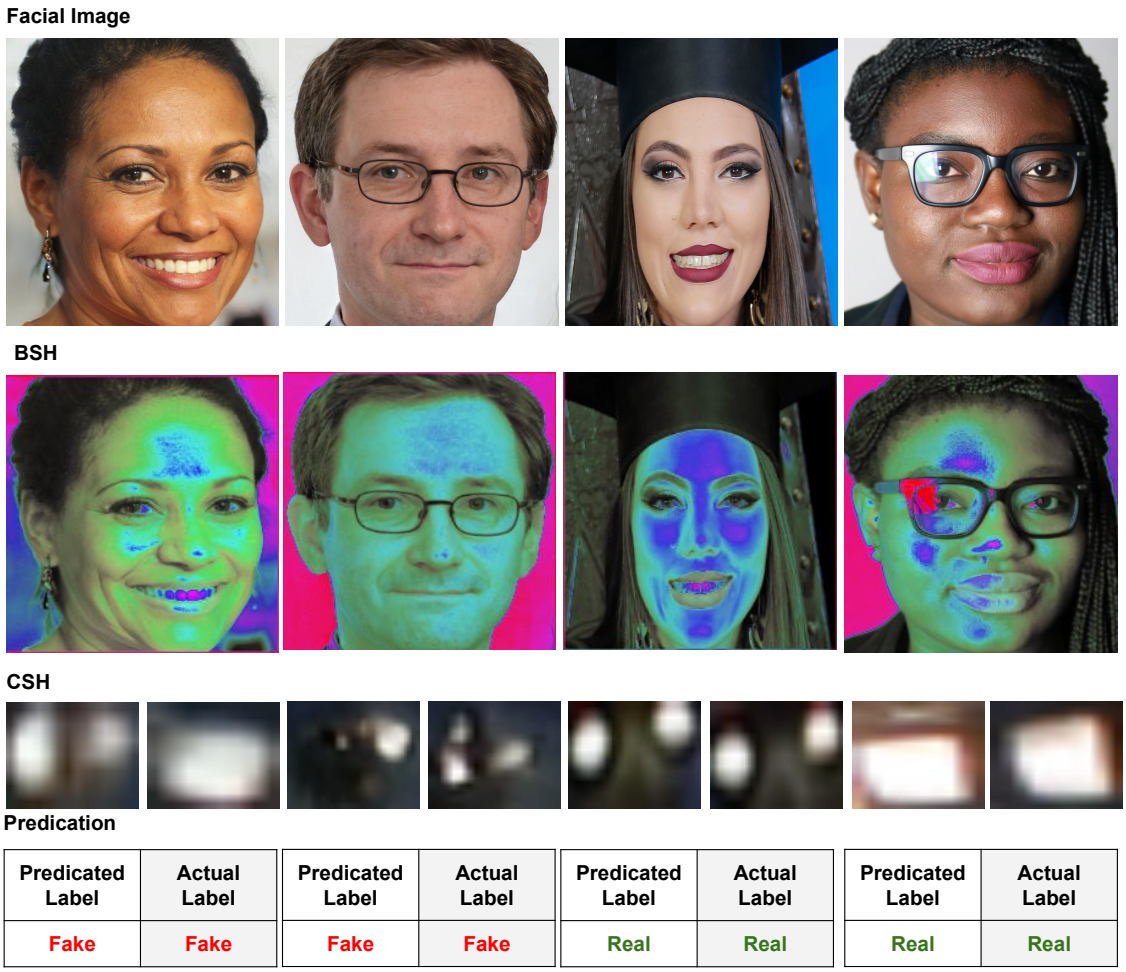
We assessed the READFake classification accuracy and compared it with other SOTA DeepFake detection methods using the READFake and FaceForensics++ datasets. We used ResNet152-V2 as the feature extraction backbone after splitting the READFake dataset with an 80:20 (training vs. validation) ratio. We trained the models on the GPU

Table 12: Accuracy comparison with SOTA methods on the READFake dataset, the FF++ dataset, and the Celeb-DF dataset. Results of some other methods are cited directly from [44].

Methods	**READFake Accu 1**	**FF++ Accu 2**	**Celeb-DF AUC(%)**
Capsule [128]	53.43%	96.60%	54.30%
SMIL [101]	-	96.8%	56.3%
Two Branch [114]	-	93.18%	73.41%
EfficientNet-B4 [39]	62.89%	99.70%	64.29%
SPSL [105]	-	96.9%	72.4%
MaDD [181]	86.93%	99.8%	67.44%
CORE [131]	88.23%	**99.97%**	79.45%
EfficientNetV2L [124]	-	99.40%	66.90%
ICT-Ref [58]	-	98.56%	**94.43%**
Chen et al. [44]	-	98.40%	67.40%
Ours (READFake)	**99.00%**	99.65%	72.67%

environment for 15 hours using the Google Colab Compute Engine (GCE) VM backend with (NVIDIA Tesla-P100-PCIE-16GB) model for 352 iterations with RMSprop optimizer, batch size of 8, and learning rate of 1e-5.

We tested Capsule [128], EfficientNet-B4 [39], MaDD [181], and CORE [131] methods under the cross-dataset setup to evaluate their generalization using the READFake testing dataset. Also, we checked the FaceForensics++ training and testing datasets as a reference for evaluating all SOTA methods with the proposed READFake classification method. The results in Table 12 (Accu 1) demonstrate that READFake achieves 99.00% accuracy with the READFake testing datasets, which outperforms EfficientNet-B4 [38, 39] (62.89%), Capsule [127, 128] (53.43%), and MaDD [181] (86.93%) with the READFake testing dataset and FaceForensics++ training dataset. The results in Table 12 (Accu 2) show that the READFake classification method on the FaceForensics++ training



Predicated Label	Actual Label	Predicated Label	Actual Label	Predicated Label	Actual Label	Predicated Label	Actual Label
Fake	Fake	Fake	Fake	Real	Real	Real	Real

Figure 31: Samples of the READFake testing dataset classification results.

and testing datasets achieves 99.65% accuracy, which is as good as EfficientNet-B4 [39] (99.70%), MaDD [181] (99.80%), and CORE [131] (99.97%), and much better than Two Branch [114] (93.18%) and Capsule [128] (96.60%). The results show that the READ-Fake classification method works well on both datasets. Figure 31 presents samples of the READFake and FaceForensics++ testing dataset classification results. READFake can discern DeepFake images well on realistic human facial images.

9.3.4 Cross-dataset Evaluation

Table 13: Comparison of READFake and SOTA methods with the DFDC dataset. Results of some other methods are cited directly from [12].

Method	**Log loss**
Good At Curve Fitting [12]	**0.192**
Deep Diggers [12]	0.199
WestLake [12]	0.200
Selim Seferbekov [12]	0.203
Vladislav Leketush [12]	0.228
Ours (READFake)	0.198

We conducted a cross-dataset evaluation with the Celeb-DF and DFDC datasets. The objective is to assess the transferability of READFake in detecting DeepFake with various public datasets. The testing results of READFake with the Celeb-DF and DFDC datasets are compared with the SOTA DeepFake detection methods in the Area Under the ROC Curve (AUC) and log loss scores. The results in Tables 12 (AUC) show that the READFake AUC score is (72.67%) with the Celeb-DF testing dataset, which is the fourth-best AUC score of all the comparing SOTA methods, behind ICT-Ref [58], CORE [131], and Two Branch [114]. However, the ICT-Ref [58] and Two Branch [114] performances

with the FaceForensics++ datasets were much lower than READFake (99.65%). The results in Table 13 demonstrate that the READFake's log loss score with the DFDC dataset is only (0.198), which is the second-best of all compared methods with the DFDC dataset, slightly behind Good At Curve Fitting [12] (0.192). The results demonstrate that the proposed READFake classification method could outperform as a stand-alone detector and enhance the existing methods as a complementary module.

9.3.5 Classification Using Different Backbone Models for Feature Extraction

Table 14: Classification performance comparison on READFake dataset with different backbone models for feature extraction.

Backbone	**Accuracy**	**Loss**
READFake (Inception-V3)	96.40%	0.177
READFake (EfficientNet-B1)	97.00%	0.109
READFake (DenseNet-169)	98.80%	0.048
READFake (ResNet152-V2)	**99.00%**	**0.011**

We evaluated the READFake method with four different neural network backbones for feature extraction, including ResNet152-V2, DenseNet-169, EfficientNet-B1, and Inception-V3. Table 14 shows the classification accuracy and loss with the READFake training and testing datasets. Overall, READFake performs well with different feature extractors. For example, the worst accuracy is 96.40% with READFake (Inception-V3), and the highest loss is 0.177 with READFake (Inception-V3). READFake (ResNet152-V2) is the best in both accuracy (99.00%) and loss (0.011).

Table 15: Classification performance comparison with READFake dataset using modular feature classifiers (i.e., CDS, EDS, CSH) for (Accu1 and Loss1: READFake (DenseNet-169), Accu2 and Loss2: READFake (ResNet152-V2)).

Feature Classifiers	**Accu1**	**Loss1**	**Accu2**	**Loss2**
CDS + EDS	83.64%	1.322	84.58%	0.920
CSH	85.05%	1.067	86.92%	0.914
BSH	88.79%	0.657	89.72%	0.627
CDS + EDS + FIEP	88.20%	0.642	89.71%	0.482
CSH + FIEP	89.61%	0.523	92.05%	0.433
BSH + FIEP	93.35%	0.415	94.85%	0.314
BSH + *CSH* + CDS + EDS + FIEP	**98.80%**	**0.048**	**99.00%**	**0.011**

9.3.6 Classification Using Different Feature Classifiers

We evaluated the contribution of different feature classifiers by combining various modular components (CDS, EDS, CSH, BSH, and FIEP). We tested it using two top-performing feature extractors, READFake (DenseNet-169) and READFake (ResNet152-V2), with the READFake dataset. Table 15 shows that the best performance can be accomplished by using all available feature classifiers of reflections and environmental factors (*BSH* + *CSH* + CDS + EDS + FIEP) on the ResNet152-V2 (99.00% in accuracy and 0.011 in loss) feature extraction. It demonstrates that the BSH feature achieves high accuracy and low loss (89.72% accuracy and 0.627 loss with ResNet152-V2) compared with other single components such as CDS, EDS, and CSH. Also, using FIEP over BSH improves the accuracy from 89.72% to 94.85% and reduces the loss from 0.627 to 0.314. The results indicate that using a single classifier alone is not a good idea, and combining various modular classifiers can greatly improve performance.

9.4 Summary

We proposed a novel **R**eflection and **E**nvironment-**A**ware **D**eep**Fake (READ-Fake)** detection technique that effectively exploits crucial and multiple factors of an image. We implemented a DeepFake detection module that extracts various features from the specular highlights of diverse body and facial parts, such as color components, shapes, and textures, to check the coordination with the surrounding environmental factors. We verified the hypothesis that the existing DeepFake creation methods are unsuccessful in harmonizing their counterfeits with the reflective and surrounding components. We also created a new READFake dataset and made it available to the research community. We conducted extensive experiments to evaluate the performance of our method using various input parameters and advanced Deep Neural Network (DNN) architectures on multiple public DeepFake datasets. Our proposed READFake achieves better accuracy (99.0%) than the SOTA methods. The modular design of READFake components makes itself use with other DeepFake detection approaches in a complementary manner.

PART 4

MASTER FACE DICTIONARY ATTACKS DETECTION

CHAPTER 10

AN OVERVIEW OF MASTER FACE DICTIONARY ATTACKS

10.1 Understanding Master Face Dictionary Attacks (MFDA)

A dictionary attack is a type of cyberattack that involves trying a large number of possible passwords or passphrases against a target system in an attempt to guess the correct one. In the case of FRS, a dictionary attack could involve trying several different faces in an attempt to bypass the FRS. A master face is a face image that passes face-based identity authentication for a large portion of the population. Master face can be used to impersonate, with a high probability of success, any user without having access to their information [129, 154]. Master face dictionary attacks (MFDA) are a type of attack that uses a pre-generated set of master faces to try to impersonate users on FRSs. The attacker creates a set of master faces by generating a large number of face images and then optimizing them to be as similar as possible to the faces of real people. Once the master faces have been created, the attacker can then try to use them to impersonate users on any FRS by presenting one of the master faces to the system and claiming to be the corresponding user. MFDA are a serious threat to the security of FRSs. They can be used to bypass FRSs that are not adequately secured, and they can be used to impersonate users in a variety of contexts, such as accessing secure websites or buildings [129, 154]. MFDA are particularly effective against FRS applications with the federated learning environment on the edge and mobile devices due to the lack of computationally

effective master face detectors.

10.2 The Technology Behind Master Face Dictionary Attacks (MFDA)

MFDA uses the power of deep learning models to create a set of master face images that never existed in the first place. Here we discuss two core advancements behind MFDA, namely generative adversarial networks (GANs) and latent variable evaluation (LVE) strategies, and how they further enable the generation of MFDA.

Deep learning is machine learning that applies neural networks to analyze datasets and look for patterns with the help of neural networks. These neural networks mimic how human brains work to learn more effectively from the data provided. Deep learning technology, paired with the availability of large face image datasets and efficient LVE strategies to train the generative models, has allowed for rapid improvement of the generation of MFDA [154].

Deep learning algorithms use neural networks to find patterns in data. Therefore, the availability of large face image datasets is essential for a good MFDA generation system. It needs examples to learn what the result in master face image looks like. It will try to discover patterns in the available face image datasets and thus extract what features are important and how they relate. That will allow it to construct a complete and convincing master face image. The result may be more or less realistic depending on the quality of the available face images in the datasets and the factors the algorithm uses.

The adaptation of GANs and LVE strategies made a significant leap in the quality

of master face images. A GAN works with two competing models: generative and discriminating. The generative model creates master face images based on the available face images in the training dataset, trying to capture the data as closely as possible to create master face images that most closely mimic the real examples in the training dataset.

A discriminative model, which could be any FRS, then tests the results of the generative model by measuring the similarity between the generated face images and the real face images. The similarity between the generated face images and the real face images can be measured using a variety of metrics. Some common metrics include the Euclidean distance, the cosine similarity, and the structural similarity index (SSIM).

The LVE algorithm uses these scores to generate new latent vectors, and the models continuously improve until the generated master faces can have a better generalization. This powerful method both simplifies the learning process, making it more accessible and also improves the outcome by incorporating a mechanism designed to minimize the chance that its generated master face images would be discriminated from authentic ones by the FRS [154].

CHAPTER 11

MASTER FACE DICTIONARY ATTACKS DETECTION LITERATURE REVIEW

In this chapter, first, we briefly present the current facial image generation techniques. Next, we review notably related facial recognition systems. Lastly, we discuss existing master face attacks and latent variable evaluation strategies.

11.1 Facial Image Generation

Today's most popular application of image generation is the generation of highly realistic faces by learning the latent visual spaces and sampling from them to generate a new facial image interpolated from real ones. There are two main techniques for the generation of highly realistic faces, using Variational Autoencoders (VAEs) [93, 141] or using Generative Adversarial Networks (GANs) [66].

VAE consists of three main components: an encoder, a decoder, and a loss function. In order to generate an image with VAE, the encoder maps the input image into a latent lower-dimensional representation space. The decoder then gets as input the latent representation of the input image and outputs the probability distribution for each of the pixels in the image. The VAE's loss function is the negative log-likelihood with a regularizer. The loss function is a key component of VAEs since it encourages the encoder to be informative and stochastic in generating new images with high similarity to real ones. Earlier versions of VAEs tended to generate small, low-quality, blurry images.

However, recently improved versions of VAEs [82, 140] have been developed to generate high-resolution images.

Alternatively, GANs can generate highly realistic synthetic images by driving the generated images to be statistically almost indistinguishable from real ones. Therefore, a GAN model consists of a generator (decoder) network and a discriminator (adversary) network. The generator network takes as input a random point in the latent space and decodes it into a synthetic image. The discriminator network takes an image that could be real or synthetic as input. Then, the discriminator predicates whether the input image is real from the training dataset or is synthesized by the generator network. The main objective is to fool the discriminator network by training the generator network to generate new realistic images which the discriminator fails to distinguish from real ones. For example, multiple GAN models (e.g., StyleGAN [87], StyleGAN2 [88], StyleGAN3 [86], InterFaceGAN [153], Image2StyleGAN++ [29]) have been developed to generate fake facial images with various characteristics, such as ages, expressions, backgrounds, and viewing angles.

11.2 Facial Recognition Systems (FRS)

FRS is a technology that utilizes artificial intelligence algorithms to identify individuals based on their facial features and attributes. This technology is one of the most demanded identification solutions for identity verification. It has constantly been evolving and being applied to new and innovative use cases, such as security and surveillance, electronic device unlocking, and controlling access to secure areas.

Large publicly available facial datasets are essential for the continuous advancement and improvement of facial recognition technology. Today, various large facial datasets are commonly used by researchers and developers in the field of computer vision and facial recognition to train and evaluate their algorithms, including the Labeled Faces in the Wild (LFW) dataset [81], the MegaFace dataset [90], the WIDER Face dataset [177], the CelebFaces Attributes (CelebA) dataset [108], the CASIA-WebFace dataset [179], and MS-Celeb dataset [70].

Furthermore, several cutting-edge face recognition models have dominated the field for the past few years, such as VGG-Face [135], FaceNet [148], OpenFace [37], DeepFace [163], DeepID [160], Dlib [92], and ArcFace [53]. In addition, various similarity matrices (e.g., cosine distance, euclidean distance, etc.) have been used to find the similarity between the learned embedding face representations. Moreover, different loss functions have been used in training face recognition models, including triplet, angular, and contrastive loss functions. Further, the performance of face recognition models can be assessed using different evaluation matrices, such as accuracy, false acceptance rate, false rejection rate, and equal error rate.

Recent studies [144, 154, 176] showed that FRS are known to be vulnerable to several types of adversarial attacks, such as inference attacks, reconstruction attacks, membership attacks, presentation attacks, and master face attacks, which cause significant challenge to their usability as they can not assess real user physical presence in unsupervised environments.

11.3 Master Face Dictionary Attacks (MFDA) Generation

MFDA refers to a machine learning attack aimed to create a master or reference face that can be used to spoof FRS. Such an attack generates a synthesized face image similar to a large portion of a facial dataset population, fooling the FRS into misidentifying the person in the generated image. [129, 154] demonstrated that it is possible to spoof FRS at high success rates using MFDA. For example, Shmelkin et al. [154] used StyleGAN [87] to generate nine master faces representing 40 percent of the 5,749 people in the Labeled Faces in the Wild (LFW) dataset [81]. Next, they used the generated master faces to spoof three different facial recognition models, including Dlib, FaceNet, and SphereFace. They have yet to test their method against commercial FRSs. However, MFDA poses a significant threat to the security and privacy of individuals, as they can be used to gain unauthorized access or steal sensitive information.

Furthermore, latent variable evaluation (LVE) strategies can be used with GAN models to evaluate the latent vector before passing it as an input into the GAN generator model in order to generate highly realistic synthetic facial images. For instance, Nguyen et al. [129] used the StyleGAN face generator and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [72] to generate high-resolution MFDA. [154] presented a comprehensive comparison between the different latent variable evaluation strategies for the MFDA generation task, such as the Limited-Memory Matrix Adaption Evaluation Strategy (LM-MA-ES) [109], the Differential Evolution (DE) [159], the LSHADE

algorithm with Rank-Based Selective Pressure Strategy (LSHADE-RSP) [156], the Improved Multi-Operator Differential Evolution algorithm (IMODE) [145], and the Nevergrad Gradient-Free Optimization algorithm (NGOpt) [106]. According to [154], LM-MA-ES performed the best among the other evaluation strategies, which indicates that it is appropriate for solving the high-dimensional black-box optimization problem.

CHAPTER 12

A COUNTERMEASURE AGAINST MASTER FACE DICTIONARY ATTACKS VIA REFLECTION-BASED IDENTIFICATION (DARI)

12.1 Background

With significant advancements in vision-based artificial intelligence (AI) technologies, Facial Recognition Systems (FRS) have emerged as one of the most practical and viable authentication approaches. The utilization of FRS has been gaining traction in various sectors, such as payment, access control, and security, due to their quick authentication processes, and contactless and uninterrupted user interaction. However, FRS's accuracy and reliability are known to be vulnerable to various adversarial attacks, such as identity theft, spoofing, and presentation attacks. Recently, Generative Adversarial Networks (GAN) generated master face dictionary attacks (MFDA) [129, 154] pose a significant risk to FRS with the reasonably high matching ratio (40%) to multiple enrolled face templates. MFDA is especially damaging to FRS applications with the federated learning environment on the edge and mobile devices due to the lack of computationally effective master face detectors. While [129] has suggested that GAN face detection methods may be able to detect MFDA, there is currently no widely accepted or implemented MFDA countermeasure available for edge and mobile applications. Hence, developing a lightweight and real-time MFDA detector optimized for the edge computing environment could significantly enhance spoofing detection capabilities, ultimately enabling the more

widespread and secure deployment of AI-based FRS in edge computing applications.

This chapter presents a novel countermeasure against Master Face **D**ictionary **A**ttacks using a **R**eflection-based **I**dentification (**DARI**) system. DARI takes specular reflections on different facial parts (e.g., eyes, cheeks, nose, chin, forehead, etc.) to extract their physiological characteristics, such as intensity and shape. We hypothesize that the existing MFDAs fail to coordinate their counterfeits with the reflective elements on each facial component and demonstrate noticeable physiological flaws on different facial parts. Instead of assessing particular facial attributes or features, we have developed a streamlined and sensible feature extraction network based on Vision Transformer (ViT) technology [59]. This network can identify incongruences between specular highlights and physiological traits by analyzing non-overlapping, minuscule segments of a facial image. Our lightweight and low-latency approach renders it an efficient and practical solution for facial recognition tasks. The proposed DARI model leverages the strengths of both Convolutional Neural Networks (CNN) and ViT architectures to incorporate and process both local and global information, ultimately improving the representation learning process from facial images with a reduced number of parameters. By fusing these two methods, DARI effectively encodes the spatially-localized features captured by CNNs with the global context awareness capabilities of ViT. As illustrated in Figure 32, DARI comprises Training Data Annotation (TDA), Face Specular Highlights Detection (FSHD), and Feature Extraction and Classification (FEC) modules. We create a new DARI dataset with high-resolution real face and master face (MF) images. The *TDA* annotates the Face Specular Highlight (FSH) regions with a range of environmental parameters to enable

more accurate and precise analysis. For a given input image, the *FSHD* module can identify various FSHs across different regions of the face by analyzing the HSV (hue, saturation, value) color space of the pixels within the image. The *FEC* module employs a lightweight ViT-based backbone model to extract features effectively from the *FSH* images. The extracted features are then used to classify the input image as either MFDA or real. We have conducted extensive experiments to evaluate DARI's performance on public GAN-face detection datasets. The empirical results show that DARI achieves very high accuracy ranging from 97.83% to 99.56% against state-of-the-art MFDA and fast detection speed (less than 11 ms) on mobile devices. Further, the modular design of DARI renders itself a complementary MFDA detection module for any existing FRS.

Our contributions include:

- Generating and annotating a new DARI dataset with MF and real face images for MFDA detection.
- Exploiting reflective elements of human face to detect physiological flaws effectively.
- Designing a lightweight, modular, and real-time approach to render a complementary MFDA detection module for edge and mobile FRSs.

12.2 Proposed Architecture

DARI consists of Training Data Annotation (TDA), Face Specular Highlights Detection (FSHD), and Feature Extraction and Classification (FEC) modules to analyze the

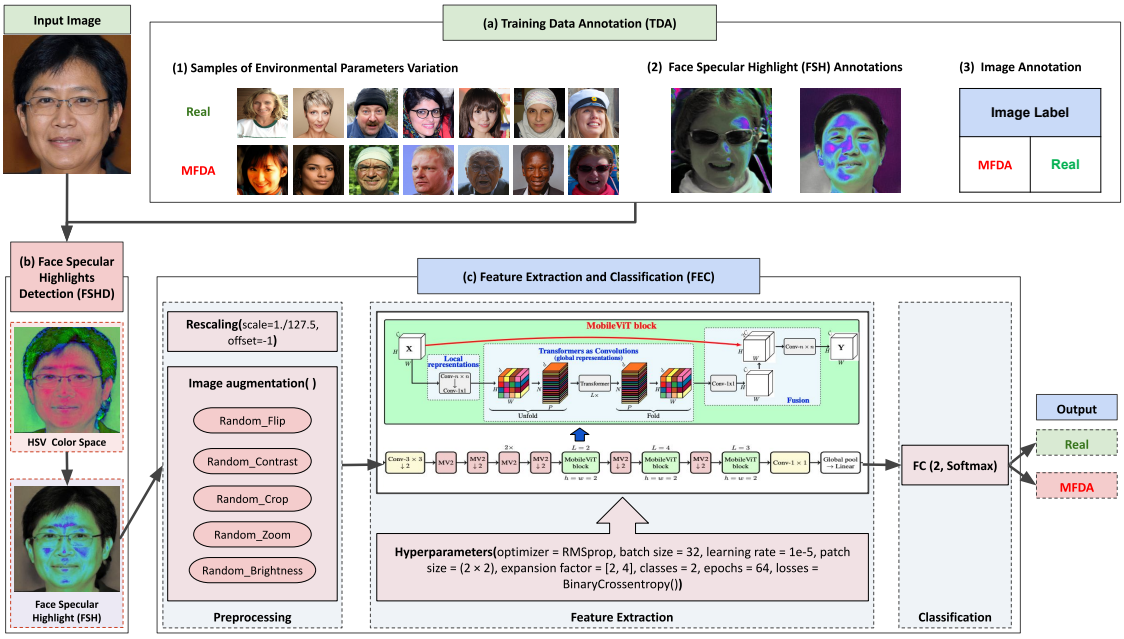


Figure 32: The DARI architecture block-diagram.

semantic aspect of the MFDAs using inconsistencies among specular highlights on various facial parts and physiological flaws of the MF images.

12.2.1 Training Data Annotation (TDA)

A large-scale benchmark dataset for evaluating GAN-based MFDA detection still needs to be improved. First, we created a DARI dataset [15] by collecting and annotating real-face and GAN-based MF images. The DARI dataset contains 28,620 (29.40 GB) high-resolution facial images. Since MFDA detection is a binary classification problem, we collected 14,310 MF images using various start-of-the-art (SOTA) GAN models, including 6,580 images from StyleGAN, 6,580 images from StyleGAN2, and 1,150 images

from StyleGAN3. In addition, we also collected 14,310 real-face images from diverse datasets, including 4,770 images from the FFHQ dataset, 4,770 images from the CelebA-HQ dataset, and 4,770 from the CelebA dataset.

Second, as presented in Figure 32 (a-1), the *TDA* module extracts environmental parameters, including illumination conditions, background colors, indoor or outdoor settings, face pose orientations, age, ethnicity, and appearances (e.g., wearing makeup and accessories) from the DARI dataset images (1). Then, TDA annotates dataset images in a couple of different types. The *Face Specular Highlight (FSH) annotation* in Figure 32 (a-2) identifies various highlight patterns from the reflective facial regions. The *image annotation* in Figure 32 (a-3) labels the images either Real or MFDA. TDA also resizes all images to the same 256×256 images. TDA applies various augmentations, including horizontal flip, crop, and adjusting brightness and saturation to increase the diversity of the training set.

12.2.2 Face Specular Highlights Detection (FSHD)

The *FSHD* module in Figure 32 (b) detects the FSH patterns from various face parts by taking a 256×256 RGB image as an input. First, it remaps the primary colors of the input images into HSV (Hue (H), Saturation (S), Value (V)) color space dimensions. *H* specifies the angle of the color from 0 to 360 degrees, *S* controls the used color amount from 0 to 100 percent, and *V* maintains the brightness of the color from 0 (black) to 100. Then, *FSHD* identifies the FSH regions by performing backward conversion from HSV to BGR (RGB, revered) to highlight the brightest point with high and low saturation values.

12.2.3 Feature Extraction and Classification (FEC)

As illustrated in Figure 32 (c), the *FEC* module conducts image preprocessing, deep hierarchical feature extraction from the *FSH* images, and classification by employing a lightweight ViT-based backbone model [116].

We built an input processing pipeline that standardizes the input images by rescaling their values from the [0, 255] range to the [-1, 1] range. Then, it applies random augmentation transforms during training, including contrast, brightness, horizontal flip, crop, and zoom.

The *FEC* also leverages the strengths of both CNN and ViT architectures by fusing the spatially-localized features captured by CNNs with the global context awareness capabilities of ViT, ultimately improving the representation learning process from facial images with fewer parameters. The *FEC* architecture is comprised of six blocks. The first block consists of a strode 3×3 standard convolution, followed by one MobileNetV2 (MV2) [146] inverted residual block. The second block contains three inverted residual MV2s for downsampling the resolution of the intermediate feature maps. The 3 to 5 blocks comprise a sequence of one inverted residual MV2 for downsampling and a MobileViT that captures local features through convolutional layers and global elements from the small patches using a transformer block [170] with different lengths. The transformer block learns a set of identity and reflection vectors from different facial parts simultaneously in a seamlessly unified model, instead of using several separate models that each is accountable for one of the face regions. The output of the 5th block is considered as the identity and reflection information which then is fed into the 6th block. The 6th

block comprises a global average pooling (GAP) and fully connected layers. The GAP layer performs downsampling, and the fully connected layer (predication layer) returns a probability distribution with two nodes and a softmax activation function for binary classification. A binary cross-entropy probabilistic loss function is used to compute the cross-entropy loss between actual and predicted labels and to measure the model's accuracy during training and testing. Eventually, it creates a binary classification result (either MF or real).

All images were pre-processed and scaled between -1 and 1. We used the Glorot normal initializer from the Keras library for the default weight initialization. We trained all three models on the GPU environment using the Google Colab Compute Engine (GCE) VM backend with (NVIDIA Tesla-P100-PCIE-16GB) model for 64 iterations with an Adam optimizer, batch size of 32, a learning rate of 1e-5, and patch size of 2×2 for the transformer blocks. In MV2s, we used an expansion factor of 4 for DARI (S) and DARI (XS), except for DARI (XXS), we used an expansion factor of 2.

12.3 Evaluations

We conducted *inference time* and *classification performance* tests on DARI modules trained on three different backbone architectures (DARI (S), DARI (XS), and DARI (XXS) in Table 16) with three distinct datasets (DARI (StyleGAN), DARI (StyleGAN2), and DARI (StyleGAN3) in Table 17). Our evaluation study aimed to determine whether particular combinations of backbones and datasets produce better inference time results on resource-constrained devices and to evaluate the effectiveness of the DARI modules in

classifying images.

Our experimental setup measured the *inference time* of different DARI backbones (size and parameters) with CPU and GPU environments. A batch of 32 images was used for GPU, while one image per batch is used for an 8-core CPU. We evaluated its *classification performance* on predefined image classes, including binary cross-entropy loss function, a dense layer of two nodes, and softmax activation at the top of every network.

Table 16: Inference time with three different DARI backbones on CPU and GPU.

Backbones	**Size (MB)**	**Params**	**Inf. Time (ms)**	
			CPU Batch Size (1)	**GPU** Batch Size (32)
DARI (S)	81.6 MB	7,040,002	10.55 ms	194 ms
DARI (XS)	32.8 MB	2,774,890	7.56 ms	146 ms
DARI (XXS)	**16 MB**	**1,306,658**	**3.93 ms**	**126 ms**

The *inference time* tests results are presented in Table 16. DARI (XXS), with 1.3 M parameters and 16 MB size, is the fastest network (within 4 ms) across all devices. On the other hand, DARI (S), with 7 M parameters and 81.6 MB size, is the slowest. All models can evaluate within 200 ms for the typical batch size of 32 images with GPU and within 11 m for a single batch with CPU. The results have important implications for developing and deploying lightweight DARI modules to detect MFDA in practical edge and mobile environments.

The outcome of *classification performance* tests, including the classification accuracy, loss, false acceptance rate (FAR), and false rejection rate (FRR), are summarized in Table 17. We observed that the DARI (S), the largest backbone, consistently outperformed the XS and XXS backbones in all metrics on DARI (StyleGAN) and DARI (StyleGAN3)

Table 17: Classification performance with three different DARI backbones and datasets.

Backbones	**DARI (StyleGAN):** A dataset with 13,160 images (50% real & 50% MF), training 10,000, validation 2,100, and testing 1,060.				**DARI (StyleGAN2):** A dataset with 13,160 images (50% real & 50% MF), training 10,000, validation 2,100, and testing 1,060.				**DARI (StyleGAN3):** A dataset with 2,300 images (50% real & 50% MF), training 1,600, validation 500, and testing 200.			
	Acc↑	**Loss↓**	**FAR↓**	**FRR↓**	**Acc↑**	**Loss↓**	**FAR↓**	**FRR↓**	**Acc↑**	**Loss↓**	**FAR↓**	**FRR↓**
DARI (S)	**99.25%**	**0.056**	**0.56%**	**0.94%**	98.87%	0.082	0.37%	1.88%	**99.56%**	**0.010**	**0.37%**	**0.50%**
DARI (XS)	98.87%	0.071	**0.56%**	1.69%	**99.34%**	**0.030**	0.37%	**0.94%**	99.43%	0.011	0.50%	0.62%
DARI (XXS)	97.83%	0.116	1.69%	2.64%	99.25%	0.032	**0.18%**	1.32%	98.81%	0.044	0.87%	1.75%

datasets. However, the smaller backbones, DARI (XS) or DARI (XXS), result in better performance on DARI (StyleGAN2) dataset. Overall, our findings indicate that the *classification performance* is very effective (e.g., higher than 97.83 % accuracy) regardless of the choice of backbone and dataset.

12.4 Summary

We proposed a novel technique against MFDA, called DARI, that generates and annotates a new dataset with MFDA and real images, to detect MFDA. It uses reflective elements to extract characteristics for MFDA detection that enables to detect noticeable physiological flaws. We designed it as a lightweight, modular, real-time system to render a complementary MFDA detection module for edge and mobile FRSs. The empirical results show that DARI achieves high detection accuracy ranging from 97.83% to 99.56% and rapid detection speed (less than 11 ms) against the current SOTA MFDAs.

PART 5

CONCLUSIONS

CHAPTER 13

CONCLUSIONS AND FUTURE WORK

In this dissertation, we introduced ML-CHIEFS: Machine Learning-based Corneal-specular Highlight Imaging for Enhancing Facial Recognition Security. Based on the hypothesis that existing facial spoofing techniques can not align their counterfeits with reflective components, we proposed countermeasures against facial biometric presentation attacks (PA), detect DeepFakes, and identify master face dictionary attacks (MFDA) using intelligent ML-based specular highlights detections.

For liveness detection against facial biometric PA, we introduced AIME, a software-based human liveness detection method for mobile device security. AIME utilizes the "Your Eyes Show What Your Eyes See! (YES2)" concept using screen display and human corneal-specular reflection as a challenge-response method for liveness authentication. We designed and built multiple ML functions to identify reflective patterns and perform authentication, including eye image acquisition, reflection image augmentation, super-resolution, feature extraction, and classification. We have also created two ML datasets for learning liveness authentication, the eye images for reflection localization and corneal reflection images for super-resolution and classification. We have built a lightweight ML package for Android, iOS, and web applications. We have shown that AIME provides an accurate and efficient PAD using only a front-facing camera and has broad applicability for various mobile and IoT device apps, either as a stand-alone liveness detection app or

as a complementary software solution for touchless biometric systems.

For DeepFakes detection, we proposed novel ML-based DeepFake detection technologies, including CHIEFS (Corneal-Specular Highlights Imaging for Enhancing Fake-Face Spotter), MobiDeep (Mobile DeepFake Detection through ML-based Corneal-Specular Backscattering), and READFake (Reflection and Environment-Aware DeepFake). CHIEFS detects various corneal-specular and facial highlights features and checks the ensemble of the highlights with the surrounding environmental factors. The experimental results show that the detection accuracy increases from 86.05% when using reflection shape similarity alone to 99.00% using ResNet-50-V2 DNN architecture. MobiDeep is a real-time, cloudless, lightweight mobile application for human visual DeepFake detection. MobiDeep achieved high accuracy (98.7%) and rapid detection speed in detecting sophisticated DeepFake images within 200 ms. The READFake detection technique uses specular highlights on various facial and body parts and environmental factors. We have performed extensive experiments to evaluate the performance of READFake using various input parameters and cutting-edge DNN architectures on multiple public DeepFake datasets. The empirical results show that READFake achieves (99.0%) accuracy in detecting DeepFake images.

We also developed a novel countermeasure against MFDA, named a Reflection-based Identification (DARI) system. DARI exploits specular highlights on various facial components and physiological characteristics of a human face, as we find the existing MFDAs fail to coordinate their counterfeits on the reflective elements. Using a lightweight and low-latency vision transformer, we built a feature extractor network to identify the

inconsistencies among specular highlights and physiological characteristics in a facial image. We conducted extensive experiments to evaluate DARI's performance on public GAN-face detection datasets and mobile devices. The empirical results show that DARI achieves high detection accuracy ranging from 97.83% to 99.56% and rapid detection speed (less than 11 ms) against SOTA MFDA.

In our future work, we will study the feasibility and integration of AIME to continuously and passively verify the user's liveness against PA on zero-trust security authentication mechanisms that repeatedly re-authenticate the user during the session, including technologies such as video conferencing and online proctoring software. Additionally, in conjunction with classifying DeepFake images, we intend to utilize reflective patterns to detect the methods used in generating DeepFakes.

APPENDIX A

LIST OF PUBLICATIONS

The following papers have been published as a direct result of the research discussed in this dissertation:

- Mohzary, M., Almalki, K., Choi, BY., Song, S. (2023). CHIEFS: Corneal-Specular Highlights Imaging for Enhancing Fake-Face Spotter. In: Jourdan, GV., Mounier, L., Adams, C., Sêdes, F., Garcia-Alfaro, J. (eds) Foundations and Practice of Security. FPS 2022. Lecture Notes in Computer Science, vol 13877. Springer, Cham. https://doi.org/10.1007/978-3-031-30122-3_10

- Mohzary, M., Almalki, K. J., Choi, B. Y., & Song, S. (2023, January). MobiDeep: Mobile DeepFake Detection through Machine Learning-based Corneal-Specular Backscattering. In 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC) (pp. 1104-1109). IEEE.

- Mohzary, M., Almalki, K. J., Choi, B. Y., & Song, S. (2022). Apple In My Eyes (AIME): Liveness Detection for Mobile Security Using Corneal Specular Reflections. IEEE Internet of Things Journal.

- Mohzary, M., Almalki, K. J., Choi, B. Y., & Song, S. (2021, June). Your Eyes Show What Your Eyes See (Y-EYES) Challenge-Response Anti-Spoofing Method for Mobile Security Using Corneal Specular Reflections. In Proceedings of the 1st

Workshop on Security and Privacy for Mobile AI (pp. 25-30).

- Mohzary, M., Almalki, K. J., Choi, B. Y., & Song, S. (2021, June). Apple in my eyes (AIME) liveness detection for mobile security using corneal specular reflections. In Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (pp. 489-490).

REFERENCE LIST

[1] All it Takes to Fool Facial Recognition at Airports and Border Crossings is a Printed Mask, Researchers Found. [Online]. Available: https://www.businessinsider.com/facial-recognition-fooled-with-mask-kneron-tests-2019-12. (Accessed on 03/01/2021).

[2] Avatarify: AI Face Animator App. [Online]. Available: https://avatarify.ai. (Accessed on 11/15/2021).

[3] CHIEFS DeepFake Detection Dataset. [Online]. Available: https://github.com/READFake/CHIEFS-DFD-Dataset. (Accessed on 11/26/2022).

[4] Contactless Biometrics Technology Market. [Online]. Available: https://www.thebrainyinsights.com/report/contactless-biometrics-technology-market-13323. (Accessed on 04/10/2023).

[5] Face Detection Concepts. [Online]. Available: https://developers.google.com/ml-kit/vision/face-detection/. (Accessed on 04/13/2021).

[6] Face Swapper. [Online]. Available: https://icons8.com/swapper. (Accessed on 10/14/2022).

[7] Faceswap: Deepfakes Software for All. [Online]. Available: https://github.com/deepfakes/faceswap. (Accessed on 10/01/2021).

[8] Guide for Setting Up Human Skin. [Online]. Available: https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/Materials/HowTo/Human_Skin. (Accessed on 03/05/2023).

[9] Hyper-Realistic Masks Offer One Way to Keep a Straight Face. [Online]. Available: https://www.japantimes.co.jp/news/2020/12/16/national/hyper-realistic-masks/. (Accessed on 03/01/2021).

[10] Importing a TensorFlow GraphDef Based Models Into TensorFlow.js. [Online]. Available: https://www.tensorflow.org/js/tutorials/conversion/import_saved_model. (Accessed on 04/13/2021).

[11] ISO/IEC 30107-3:2017 - Information Technology - Biometric Presentation Attack Detection - Part 3: Testing and Reporting. [Online]. Available: https://www.iso.org/standard/67381.html. (Accessed on: 04/13/2021).

[12] Kaggle Deepfake Detection Challenge. [Online]. Available: https://www.kaggle.com/c/deepfake-detection-challenge/leaderboard. (Accessed on 10/01/2021).

[13] KAIST Unveils Deepfake Detecting Mobile App. [Online]. Available: https://en.yna.co.kr/view/AEN20210330004200320. (Accessed on 12/01/2021).

[14] Lexica Stable Diffusion Search Engine. [Online]. Available: https://lexica.art/. (Accessed on 11/24/2022).

[15] Master Face Dictionary Attacks via Reflection-Based Identification (DARI) Dataset. [Online]. Available: https://github.com/READFake/DARI-MFDA-Detection-Dataset. (Accessed on 03/01/2023).

[16] Pentagon's Race Against Deepfakes. [Online]. Available: https://www.cnn.com/interactive/2019/01/business/pentagons-race-against-deepfakes/. (Accessed on 04/10/2023).

[17] Photo AI. [Online]. Available: https://photoai.com. (Accessed on 03/11/2023).

[18] Profile your App Performance. [Online]. Available: https://developer.android.com/studio/profile. (Accessed on 04/13/2021).

[19] READFake Dataset. [Online]. Available: github.com/READFake/READFake_Dataset. (Accessed on 11/11/2022).

[20] Reface. [Online]. Available: https://reface.app. (Accessed on 12/01/2021).

[21] Remote Identity Proofing: Attacks and Countermeasures. [Online]. Available: https://www.enisa.europa.eu/publications/remote-identity-proofing-attacks-countermeasures. (Accessed on 04/10/2023).

[22] Secure Random Generators (CSPRNG). [Online]. Available: https://cryptobook.nakov.com/secure-random-generators/secure-random-generators-csprng. (Accessed on 04/13/2021).

[23] Tackling Deepfakes in European Policy. [Online]. Available: https://www.europarl.europa.eu/stoa/en/document/EPRS_STU(2021)690039. (Accessed on 02/15/2023).

[24] These Hyper-Realistic Masks are Being Used to Train Facial Recognition Tech. [Online]. Available: https://www.theverge.com/2018/11/20/18105189. (Accessed on 01/10/2021).

[25] URME Paper Mask. [Online]. Available: http://www.urmesurveillance.com/urme-paper-mask. (Accessed on 04/13/2021).

[26] What if Deepfakes Made Us Doubt Everything we See and Hear? [Online]. Available: https://www.europarl.europa.eu/stoa/en/document/EPRS_ATA(2021)690046. (Accessed on 02/15/2023).

[27] What is FIDO? [Online]. Available: https://fidoalliance.org/what-is-fido/. (Accessed on 04/13/2023).

[28] Wombo. [Online]. Available: https://www.wombo.ai. (Accessed on 12/01/2021).

[29] Abdal, R., Qin, Y., and Wonka, P. Image2StyleGAN++: How to Edit the Embedded Images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8296–8305.

[30] Abdi, H., and Williams, L. J. Principal Component Analysis. *Wiley Interdisciplinary Reviews: Computational Statistics 2*, 4 (2010), 433–459.

[31] Ajder, H., Patrini, G., Cavalli, F., and Cullen, L. The State of Deepfakes: Landscape, Threats, and Impact. *Amsterdam: Deeptrace 27* (2019).

[32] Ali, A., Deravi, F., and Hoque, S. Directional Sensitivity of Gaze-Collinearity Features in Liveness Detection. In *2013 Fourth International Conference on Emerging Security Technologies* (2013), IEEE, pp. 8–11.

[33] AlRousan, M., Intrigila, B., et al. Multi-Factor Authentication for E-Government Services Using a Smartphone Application and Biometric Identity Verification. *Journal of Computer Science 16*, 2 (2020), 217–224.

[34] Anthony, P., Ay, B., and Aydin, G. A Review of Face Anti-Spoofing Methods for Face Recognition Systems. In *2021 International Conference on Innovations in Intelligent Systems and Applications (INISTA)* (2021), IEEE, pp. 1–9.

[35] Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., and Akin, E. A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics 12*, 6 (2023), 1333.

[36] Backes, M., Chen, T., Duermuth, M., Lensch, H. P. A., and Welk, M. Tempest in a Teapot: Compromising Reflections Revisited. In *2009 30th IEEE Symposium on Security and Privacy* (2009), pp. 315–327.

[37] Baltrušaitis, T., Robinson, P., and Morency, L.-P. Openface: An Open Source Facial Behavior Analysis Toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2016), IEEE, pp. 1–10.

[38] Bonettini, N., Cannas, E. D., Mandelli, S., Bondi, L., Bestagini, P., and Tubaro, S. GitHub: Video Face Manipulation Detection Through Ensemble of CNNs. [Online]. Available: https://github.com/polimi-ispl/icpr2020dfdc. (Accessed on 10/01/2021).

[39] Bonettini, N., Cannas, E. D., Mandelli, S., Bondi, L., Bestagini, P., and Tubaro, S. Video Face Manipulation Detection Through Ensemble of CNNs. In *2020 25th International Conference on Pattern Recognition (ICPR)* (2021), pp. 5012–5019.

[40] Bud, A. Facing the Future: The Impact of Apple FaceID. *Biometric Technology Today 2018*, 1 (2018), 5–7.

[41] Bulling, A., and Gellersen, H. Toward Mobile Eye-Based Human-Computer Interaction. *IEEE Pervasive Computing 9*, 4 (2010), 8–12.

[42] Bulling, A., Ward, J. A., Gellersen, H., and Troster, G. Eye Movement Analysis for Activity Recognition Using Electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 4 (2010), 741–753.

[43] Chen, B., Liu, X., Zheng, Y., Zhao, G., and Shi, Y.-Q. A Robust GAN-Generated Face Detection Method Based on Dual-Color Spaces and an Improved Xception. *IEEE Transactions on Circuits and Systems for Video Technology 32*, 6 (2021), 3527–3538.

[44] Chen, L., Zhang, Y., Song, Y., Liu, L., and Wang, J. Self-Supervised Learning of Adversarial Example: Towards Good Generalizations for Deepfake Detection.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18710–18719.

[45] Chollet, F. Xception: Deep Learning With Depthwise Separable Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1251–1258.

[46] Chollet, F. *Deep Learning With Python*. Simon and Schuster, 2021.

[47] Chrzan, B. M. Liveness Detection for Face Recognition. *Master's Thesis* (2014).

[48] Ciftci, U. A., Demir, I., and Yin, L. Fakecatcher: Detection of Synthetic Portrait Videos Using Biological Signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[49] Czajka, A. Pupil Dynamics for Iris Liveness Detection. *IEEE Transactions on Information Forensics and Security 10*, 4 (2015), 726–735.

[50] Dang, H., Liu, F., Stehouwer, J., Liu, X., and Jain, A. K. On the Detection of Digital Face Manipulation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 5780–5789.

[51] Demir, I., and Ciftci, U. A. Where Do Deep Fakes Look? Synthetic Face Detection via Gaze Tracking. *arXiv Preprint arXiv:2101.01165* (2021).

[52] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09* (2009).

[53] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. Arcface: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4690–4699.

[54] Denning, D. E. Why I love biometrics: It is 'Liveness,' not Secrecy, that Counts. *Information Security* (2001).

[55] Do, N.-T., Na, I.-S., and Kim, S.-H. Forensics Face Detection From GANs Using Convolutional Neural Network. *ISITC 2018* (2018), 376–379.

[56] Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., and Ferrer, C. C. The Deepfake Detection Challenge (DFDC) Dataset. *arXiv Preprint arXiv:2006.07397* (2020).

[57] Dong, C., Loy, C. C., He, K., and Tang, X. Learning a Deep Convolutional Network for Image Super-Resolution. In *European Conference on Computer Vision* (2014), Springer, pp. 184–199.

[58] Dong, X., Bao, J., Chen, D., Zhang, T., Zhang, W., Yu, N., Chen, D., Wen, F., and Guo, B. Protecting Celebrities From DeepFake With Identity Consistency Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 9468–9478.

[59] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An Image is Worth

16x16 Words: Transformers for Image Recognition at Scale. *arXiv Preprint arXiv:2010.11929* (2020).

[60] Dufour, N., and Gully, A. Contributing Data to Deepfake Detection Research. [Online]. Available: https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html. (Accessed on 10/01/2021).

[61] Dutta, A., and Zisserman, A. The VIA Annotation Software for Images, Audio and Video. In *Proceedings of the 27th ACM International Conference on Multimedia* (New York, NY, USA, 2019), MM '19, ACM.

[62] Ebihara, A. F., Sakurai, K., and Imaoka, H. Specular and Diffuse-Reflection-Based Face Spoofing Detection for Mobile Devices. In *2020 IEEE International Joint Conference on Biometrics (IJCB)* (2020), IEEE, pp. 1–10.

[63] Fallis, D. The Epistemic Threat of Deepfakes. *Philosophy & Technology 34*, 4 (2021), 623–643.

[64] Farid, H. *Photo Forensics*. MIT Press, 2016.

[65] Fatima, K., Nawaz, S., and Mehrban, S. Biometric Authentication in Health Care Sector: A Survey. In *2019 International Conference on Innovative Computing (ICIC)* (2019), pp. 1–10.

[66] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Networks, 2014.

[67] Gu, K., Zhou, Y., and Huang, T. FLNet: Landmark Driven Fetching and Learning Network for Faithful Talking Facial Animation Synthesis. *Proceedings of the AAAI Conference on Artificial Intelligence 34*, 07 (Apr. 2020), 10861–10868.

[68] Guo, H., Hu, S., Wang, X., Chang, M.-C., and Lyu, S. Eyes Tell All: Irregular Pupil Shapes Reveal GAN-Generated Faces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), IEEE, pp. 2904–2908.

[69] Guo, H., Hu, S., Wang, X., Chang, M.-C., and Lyu, S. Robust Attentive Deep Neural Network for Detecting GAN-Generated Faces. *IEEE Access 10* (2022), 32574–32583.

[70] Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. Ms-Celeb-1m: A Dataset and Benchmark for Large-Scale Face Recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14* (2016), Springer, pp. 87–102.

[71] Hammoud, R. I. *Passive Eye Monitoring: Algorithms, Applications and Experiments*. Springer Science & Business Media, 2008.

[72] Hansen, N., and Ostermeier, A. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation 9*, 2 (2001), 159–195.

[73] He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.

[74] He, K., Zhang, X., Ren, S., and Sun, J. Identity Mappings in Deep Residual Networks. In *European Conference on Computer Vision* (2016), Springer, pp. 630–645.

[75] He, L., Li, H., Liu, F., Liu, N., Sun, Z., and He, Z. Multi-Patch Convolution Neural Network for Iris Liveness Detection. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)* (2016), pp. 1–7.

[76] Hernandez-Ortega, J., Fierrez, J., Morales, A., and Tome, P. Time Analysis of Pulse-Based Face Anti-Spoofing in Visible and NIR. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2018), 657–6578.

[77] Hsu, C.-C., Zhuang, Y.-X., and Lee, C.-Y. Deep Fake Image Detection Based on Pairwise Learning. *Applied Sciences 10*, 1 (2020).

[78] Hu, J., Liao, X., Liang, J., Zhou, W., and Qin, Z. FInfer: Frame Inference-Based Deepfake Detection for High-Visual-Quality Videos. *IEEE Trans. Pattern Anal. Mach. Intell* (2022), 1–9.

[79] Hu, S., Li, Y., and Lyu, S. Exposing GAN-Generated Faces Using Inconsistent Corneal Specular Highlights. In *ICASSP 2021-2021 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP) (2021), IEEE, pp. 2500–2504.

[80] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 4700–4708.

[81] Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition* (2008).

[82] Huang, H., He, R., Sun, Z., Tan, T., et al. Introvae: Introspective Variational Autoencoders for Photographic Image Synthesis. *Advances in Neural Information Processing Systems 31* (2018).

[83] Huang, J., Rathod, V., Chow, D., Sun, C., Zhu, M., Fathi, A., and Lu, Z. Tensorflow Object Detection API. [Online]. Available: github.com/tensorflow/models/tree/master/object detection. (Accessed on 04/13/2021).

[84] Hulme, G. Touchless Authentication for the Post-COVID World. [Online]. Available: https://www.hpe.com/us/en/insights/articles/touchless-authentication-for-the-post-covid-world-2010.html. (Accessed on 03/01/2021).

[85] Jackson, A. S., Bulat, A., Argyriou, V., and Tzimiropoulos, G. Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression. *International Conference on Computer Vision* (2017).

[86] Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-Free Generative Adversarial Networks. In *Proc. NeurIPS* (2021).

[87] Karras, T., Laine, S., and Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4401–4410.

[88] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. CVPR* (2020).

[89] Kazemi, V., and Sullivan, J. One Millisecond Face Alignment With an Ensemble of Regression Trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1867–1874.

[90] Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. The Megaface Benchmark: 1 Million Faces for Recognition at Scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 4873–4882.

[91] Kietzmann, J., Lee, L. W., McCarthy, I. P., and Kietzmann, T. C. Deepfakes: Trick or Treat? *Business Horizons 63*, 2 (2020), 135–146. Artificial Intelligence and Machine Learning.

[92] King, D. E. Dlib-ML: A Machine Learning Toolkit. *The Journal of Machine Learning Research 10* (2009), 1755–1758.

[93] Kingma, D. P., and Welling, M. Auto-Encoding Variational Bayes. *arXiv Preprint arXiv:1312.6114* (2013).

[94] Korshunov, P., and Marcel, S. Deepfakes:A New Threat to Face Recognition? Assessment and Detection. *arXiv Preprint arXiv:1812.08685* (2018).

[95] Korshunova, I., Shi, W., Dambre, J., and Theis, L. Fast Face-Swap Using Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 3677–3685.

[96] Kumar, R., Sotelo, J., Kumar, K., de Brebisson, A., and Bengio, Y. ObamaNet: Photo-Realistic Lip-Sync From Text, 2017.

[97] Lander, C., Krüger, A., and Löchtefeld, M. " The Story of Life is Quicker Than the Blink of an Eye" Using Corneal Imaging for Life Logging. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (2016), pp. 1686–1695.

[98] Le, B. M., and Woo, S. S. Exploring the Asynchronous of the Frequency Spectra of GAN-Generated Facial Images. *CoRR abs/2112.08050* (2021).

[99] Li, H., Li, B., Tan, S., and Huang, J. Identification of Deep Network Generated Images Using Disparities in Color Components. *Signal Processing 174* (2020), 107616.

[100] Li, L., Bao, J., Yang, H., Chen, D., and Wen, F. FaceShifter: Towards High Fidelity and Occlusion Aware Face Swapping, 2020.

[101] Li, X., Lang, Y., Chen, Y., Mao, X., He, Y., Wang, S., Xue, H., and Lu, Q. Sharp Multiple Instance Learning for Deepfake Video Detection. In *Proceedings of the 28th ACM International Conference on Multimedia* (2020), pp. 1864–1872.

[102] Li, Y., Chang, M.-C., and Lyu, S. In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking. *arXiv Preprint arXiv:1806.02877* (2018).

[103] Li, Y., Yang, X., Sun, P., Qi, H., and Lyu, S. Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 3204–3213.

[104] Liu, H., Li, X., Zhou, W., Chen, Y., He, Y., Xue, H., Zhang, W., and Yu, N. Spatial-Phase Shallow Learning: Rethinking Face Forgery Detection in Frequency Domain, 2021.

[105] Liu, H., Li, X., Zhou, W., Chen, Y., He, Y., Xue, H., Zhang, W., and Yu, N. Spatial-Phase Shallow Learning: Rethinking Face Forgery Detection in Frequency Domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 772–781.

[106] Liu, J., Moreau, A., Preuss, M., Rapin, J., Roziere, B., Teytaud, F., and Teytaud, O. Versatile Black-Box Optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (2020), pp. 620–628.

[107] Liu, S., Yuen, P. C., Zhang, S., and Zhao, G. 3D Mask Face Anti-Spoofing with Remote Photoplethysmography. In *Computer Vision – ECCV 2016* (Cham, 2016), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, pp. 85–100.

[108] Liu, Z., Luo, P., Wang, X., and Tang, X. Large-scale Celebfaces Attributes (CelebA) Dataset. *Retrieved August 15*, 2018 (2018), 11.

[109] Loshchilov, I., Glasmachers, T., and Beyer, H.-G. Limited-Memory Matrix Adaptation for Large Scale Black-Box Optimization. *arXiv Preprint arXiv:1705.06693* (2017).

[110] Määttä, J., Hadid, A., and Pietikäinen, M. Face Spoofing Detection From Single Images Using Micro-Texture Analysis. In *2011 International Joint Conference on Biometrics (IJCB)* (2011), IEEE, pp. 1–7.

[111] Mansourifar, H., and Shi, W. One-Shot GAN Generated Fake Face Detection. *arXiv Preprint arXiv:2003.12244* (2020).

[112] Marra, F., Saltori, C., Boato, G., and Verdoliva, L. Incremental Learning for the Detection and Classification of GAN-Generated Images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)* (2019), IEEE, pp. 1–6.

[113] Martin, B. Detecting Facial Liveliness, Mar. 27 2018. US Patent 9,928,603.

[114] Masi, I., Killekar, A., Mascarenhas, R. M., Gurudatt, S. P., and AbdAlmageed, W. Two-Branch Recurrent Network for Isolating Deepfakes in Videos. In *European Conference on Computer Vision* (2020), Springer, pp. 667–684.

[115] Matern, F., Riess, C., and Stamminger, M. Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)* (2019), IEEE, pp. 83–92.

[116] Mehta, S., and Rastegari, M. MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. In *International Conference on Learning Representations*.

[117] Minaee, S., Abdolrashidi, A., Su, H., Bennamoun, M., and Zhang, D. Biometric Recognition Using Deep Learning: A Survey. *arXiv Preprint arXiv:1912.00271* (2019).

[118] Mirsky, Y., and Lee, W. The Creation and Detection of Deepfakes: A Survey. *ACM Computing Surveys (CSUR) 54*, 1 (2021), 1–41.

[119] Mohzary, M., Almalki, K., Choi, B.-Y., and Song, S. CHIEFS: Corneal-Specular Highlights Imaging for Enhancing Fake-Face Spotter. In *Foundations and Practice of Security* (Cham, 2023), G.-V. Jourdan, L. Mounier, C. Adams, F. Sèdes, and J. Garcia-Alfaro, Eds., Springer Nature Switzerland, pp. 158–172.

[120] Mohzary, M., Almalki, K. J., Choi, B.-Y., and Song, S. Apple In My Eyes (AIME) Liveness Detection for Mobile Security Using Corneal Specular Reflections. In

Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (2021), pp. 489–490.

[121] Mohzary, M., Almalki, K. J., Choi, B.-Y., and Song, S. Your Eyes Show What Your Eyes See (Y-EYES) Challenge-Response Anti-Spoofing Method for Mobile Security Using Corneal Specular Reflections. In *Proceedings of the 1st Workshop on Security and Privacy for Mobile AI* (2021), pp. 25–30.

[122] Mohzary, M., Almalki, K. J., Choi, B.-Y., and Song, S. Apple In My Eyes (AIME): Liveness Detection for Mobile Security Using Corneal Specular Reflections. *IEEE Internet of Things Journal* (2022).

[123] Mohzary, M., Almalki, K. J., Choi, B.-Y., and Song, S. MobiDeep: Mobile DeepFake Detection through Machine Learning-Based Corneal-Specular Backscattering. In *2023 IEEE 20th Consumer Communications Networking Conference (CCNC)* (2023), pp. 1104–1109.

[124] Nadimpalli, A. V., and Rattani, A. On Improving Cross-Dataset Generalization of Deepfake Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 91–99.

[125] Nagano, K., Seo, J., Xing, J., Wei, L., Li, Z., Saito, S., Agarwal, A., Fursund, J., Li, H., Roberts, R., et al. paGAN: Real-Time Avatars Using Dynamic Textures. *ACM Trans. Graph. 37*, 6 (2018), 258–1.

[126] Nguyen, H. H., Fang, F., Yamagishi, J., and Echizen, I. Multi-Task Learning for Detecting and Segmenting Manipulated Facial Images and Videos. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)* (2019), pp. 1–8.

[127] Nguyen, H. H., Yamagishi, J., and Echizen, I. Implementation of the Capsule Forensics v2. [Online]. Available: https://github.com/nii-yamagishilab/Capsule-Forensics-v2. (Accessed on 10/01/2021).

[128] Nguyen, H. H., Yamagishi, J., and Echizen, I. Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), IEEE, pp. 2307–2311.

[129] Nguyen, H. H., Yamagishi, J., Echizen, I., and Marcel, S. Generating Master Faces for Use in Performing Wolf Attacks on Face Recognition Systems. In *2020 IEEE International Joint Conference on Biometrics (IJCB)* (2020), IEEE, pp. 1–10.

[130] Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. T., and Nahavandi, S. Deep Learning for Deepfakes Creation and Detection: A Survey. *arXiv Preprint arXiv:1909.11573* (2019).

[131] Ni, Y., Meng, D., Yu, C., Quan, C., Ren, D., and Zhao, Y. CORE: Consistent Representation Learning for Face Forgery Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 12–21.

[132] Nightingale, S. J., and Farid, H. AI-Synthesized Faces are Indistinguishable From Real Faces and More Trustworthy. *Proceedings of the National Academy of Sciences 119*, 8 (2022), e2120481119.

[133] Nirkin, Y., Keller, Y., and Hassner, T. FSGAN: Subject Agnostic Face Swapping and Reenactment. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 7184–7193.

[134] Nishino, K. The World in an Eye [Eye Image Interpretation]. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (2004), vol. 1, IEEE, pp. I–I.

[135] Parkhi, O. M., Vedaldi, A., and Zisserman, A. Deep Face Recognition. In *British Machine Vision Conference* (2015).

[136] Perov, I., Gao, D., Chervoniy, N., Liu, K., Marangonda, S., Umê, C., Dpfks, M., Facenheim, C. S., RP, L., Jiang, J., Zhang, S., Wu, P., Zhou, B., and Zhang, W. Deepfacelab: A Simple, Flexible and Extensible Face Swapping Framework.

[137] Raghavendra, R., Raja, K. B., and Busch, C. Presentation Attack Detection for Face Recognition Using Light Field Camera. *IEEE Transactions on Image Processing 24*, 3 (2015), 1060–1075.

[138] Ramachandra, R., and Busch, C. Presentation Attack Detection Methods for Face Recognition Systems: A Comprehensive Survey. *ACM Computing Surveys (CSUR) 50*, 1 (2017), 1–37.

[139] Rathgeb, C., Dantcheva, A., and Busch, C. Impact and Detection of Facial Beautification in Face Recognition: An Overview. *IEEE Access 7* (2019), 152667–152678.

[140] Razavi, A., Van den Oord, A., and Vinyals, O. Generating Diverse High-Fidelity Images With VQ-VAE-2. *Advances in Neural Information Processing Systems 32* (2019).

[141] Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning* (2014), PMLR, pp. 1278–1286.

[142] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Nießner, M. Faceforensics++: Learning to Detect Manipulated Facial Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 1–11.

[143] Saikia, P., Dholaria, D., Yadav, P., Patel, V., and Roy, M. A Hybrid CNN-LSTM Model for Video Deepfake Detection by Leveraging Optical Flow Features. In *2022 International Joint Conference on Neural Networks (IJCNN)* (2022), IEEE, pp. 1–7.

[144] Salem, A. M. G., Bhattacharyya, A., Backes, M., Fritz, M., and Zhang, Y. Updates-Leak: Data Set Inference and Reconstruction Attacks in Online Learning. In *29th USENIX Security Symposium* (2020), USENIX, pp. 1291–1308.

[145] Sallam, K. M., Elsayed, S. M., Chakrabortty, R. K., and Ryan, M. J. Improved Multi-Operator Differential Evolution Algorithm for Solving Unconstrained Problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)* (2020), IEEE, pp. 1–8.

[146] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 4510–4520.

[147] Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. New Support Vector Algorithms. *Neural Computation 12*, 5 (2000), 1207–1245.

[148] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A Unified Embedding for Face Recognition and Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 815–823.

[149] Schuckers, S. Presentations and Attacks, and Spoofs, Oh My. *Image and Vision Computing 55* (2016), 26 – 30.

[150] Schuckers, S., Cannon, G., Tabassi, E., Karlsson, M., and Newton, E. FIDO Biometrics Requirements. *Population 5* (2019), 2–1.

[151] Shao, R., Lan, X., and Yuen, P. C. Deep Convolutional Dynamic Texture Learning With Adaptive Channel-Discriminability for 3D Mask Face Anti-Spoofing. In *2017 IEEE International Joint Conference on Biometrics (IJCB)* (2017), IEEE, pp. 748–755.

[152] Shao, R., Lan, X., and Yuen, P. C. Joint Discriminative Learning of Deep Dynamic Textures for 3D Mask Face Anti-Spoofing. *IEEE Transactions on Information Forensics and Security 14*, 4 (2018), 923–938.

[153] Shen, Y., Yang, C., Tang, X., and Zhou, B. InterfaceGAN: Interpreting the Disentangled Face Representation Learned by GANS. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 4 (2020), 2004–2018.

[154] Shmelkin, R., Friedlander, T., and Wolf, L. Generating Master Faces for Dictionary Attacks With a Network-Assisted Latent Space Evolution. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)* (2021), IEEE, pp. 01–08.

[155] Simonyan, K., and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv Preprint arXiv:1409.1556* (2014).

[156] Stanovov, V., Akhmedova, S., and Semenkin, E. LSHADE Algorithm With Rank-Based Selective Pressure Strategy for Solving CEC 2017 Benchmark Problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (2018), IEEE, pp. 1–8.

[157] Steil, J., and Bulling, A. Discovery of Everyday Human Activities from Long-Term Visual Behaviour Using Topic Models. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (New York, NY, USA, 2015), UbiComp '15, Association for Computing Machinery, p. 75â85.

[158] Steiner, H., Kolb, A., and Jung, N. Reliable Face Anti-Spoofing Using Multispectral Swir Imaging. In *2016 International Conference on Biometrics (ICB)* (2016), IEEE, pp. 1–8.

[159] Storn, R., and Price, K. Differential Evolution-a Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. *Journal of Global Optimization 11*, 4 (1997), 341.

[160] Sun, Y., Wang, X., and Tang, X. Deep Learning Face Representation From Predicting 10,000 Classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1891–1898.

[161] Sun, Z., Han, Y., Hua, Z., Ruan, N., and Jia, W. Improving the Efficiency and Robustness of Deepfakes Detection Through Precise Geometric Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 3609–3618.

[162] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2818–2826.

[163] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. Deepface: Closing the Gap to Human-Level Performance in Face Verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1701–1708.

[164] Tan, M., and Le, Q. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning* (2019), PMLR, pp. 6105–6114.

[165] Tan, R. T. *Specularity, Specular Reflectance*. Springer US, Boston, MA, 2014, pp. 750–752.

[166] Tariq, S., Lee, S., Kim, H., Shin, Y., and Woo, S. S. Detecting Both Machine and Human Created Fake Face Images in the Wild. In *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security* (2018), pp. 81–87.

[167] Tussy, K., Wojewidka, J., and Rose, J. Biometric Liveness Detection Explained. [Online]. Available: https://www.liveness.com. (Accessed on 04/01/2021).

[168] Vakhshiteh, F., Nickabadi, A., and Ramachandra, R. Adversarial Attacks Against Face Recognition: A Comprehensive Study. *IEEE Access 9* (2021), 92735–92756.

[169] Vasiljevic, I., Kolkin, N. I., Zhang, S., Luo, R., Wang, H., Dai, F. Z., Daniele, A. F., Mostajabi, M., Basart, S., Walter, M. R., and Shakhnarovich, G. DIODE: A Dense Indoor and Outdoor Depth Dataset. *CoRR abs/1908.00463* (2019).

[170] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is All you Need. *Advances in Neural Information Processing Systems 30* (2017).

[171] Viola, P., and Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conference*

on Computer Vision and Pattern Recognition. CVPR 2001 (2001), vol. 1, IEEE, pp. I–I.

[172] Wang, Y., Chen, S., Li, W., Huang, D., and Wang, Y. Face Anti-Spoofing to 3D Masks by Combining Texture and Geometry Features. In *Chinese Conference on Biometric Recognition* (2018), Springer, pp. 399–408.

[173] Wang, Y., Hao, X., Hou, Y., and Guo, C. A New Multispectral Method for Face Liveness Detection. In *2013 2nd IAPR Asian Conference on Pattern Recognition* (2013), IEEE, pp. 922–926.

[174] Wei, X., Pu, B., Lu, J., and Wu, B. Physically Adversarial Attacks and Defenses in Computer Vision: A Survey. *arXiv Preprint arXiv:2211.01671* (2022).

[175] Woo, S., et al. ADD: Frequency Attention and Multi-View Based Knowledge Distillation to Detect Low-Quality Compressed Deepfake Images. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2022), vol. 36, pp. 122–130.

[176] Xu, Y., Price, T., Frahm, J.-M., and Monrose, F. Virtual U: Defeating Face Liveness Detection by Building Virtual Models From your Public Photos. In *25th USENIX Security Symposium (USENIX Security 16)* (2016), pp. 497–512.

[177] Yang, S., Luo, P., Loy, C.-C., and Tang, X. Wider Face: A Face Detection Benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 5525–5533.

[178] Yang, X., Li, Y., and Lyu, S. Exposing Deep Fakes Using Inconsistent Head Poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), IEEE, pp. 8261–8265.

[179] Yi, D., Lei, Z., Liao, S., and Li, S. Z. Learning Face Representation From Scratch. *arXiv Preprint arXiv:1411.7923* (2014).

[180] Zhang, Z., Yi, D., Lei, Z., and Li, S. Z. Face Liveness Detection by Learning Multispectral Reflectance Distributions. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)* (2011), IEEE, pp. 436–441.

[181] Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., and Yu, N. Multi-Attentional Deepfake Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 2185–2194.

[182] Zulfiqar, M., Syed, F., Khan, M. J., and Khurshid, K. Deep Face Recognition for Biometric Authentication. In *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)* (2019), pp. 1–6.

VITA

Muhammad Ali H Mohzary was born in 1990 in Jazan Province, Saudi Arabia. He holds a master's degree in Computer Science from Kent State University, Kent, Ohio, USA. In addition, he earned a bachelor's degree in Computer Science in 2008 from Jazan University, Jazan Province, Saudi Arabia. He is a lecturer in the Computer Science Department at Jazan University. He joined the interdisciplinary Ph.D. program at the University of Missouri-Kansas City (UMKC) in 2019. His primary discipline is Computer Science, and his co-discipline is Computer Networking and Communication Systems. His research interests lie in the broad areas of Big Data Analytics and Privacy, including Machine Learning, Deep Learning, Computer Vision, and Privacy-Enhancing Technologies.

During his studies at UMKC, Muhammad published several papers in world-ranked conferences and journals, including the International Symposium on Foundations and Practice of Security (FPS), the IFIP/IEEE International Symposium on Integrated Network Management (IM), the Annual International Conference on Mobile Systems, Applications, and Services (MobiSys), the IEEE International Smart Cities Conference (ISC2), the IEEE Global Communications Conference (GLOBECOM), the IEEE Consumer Communications and Networking Conference (IEEE CCNC), and the IEEE Internet of Things Journal (IoT-J).

Furthermore, Muhammad also received several awards and scholarships during his Ph.D. at UMKC, including the Ayinala award for excellence in research, the Missouri Small Business Development Center (SBDC) tech venture ELEVATION lab scholarship,

the Comeback Kansas City ventures fellowship fund, the IEEE ComSoc student travel grant award, the UMKC graduate student travel grant award, the Balaji Krithikaivasan memorial travel scholarship, the CANSec'22 National Science Foundation (NSF) student travel grant award, the ACM MobiSys 2021 student travel grant, and the IM 2021 NSF student travel grant.