An Industry-Oriented Mini Project Report

On

"DEEP REPRESENTATIONS OF FACE AND FINGERPRINT SPOOFING"

Submitted in Partial Fulfillment of the Academic Requirement for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

Electronics and Communication Engineering

Submitted By:

D.SAKETH	**(20R01A04K5)**
G.VINAY	**(20R01A04K9)**
G.VAISHNAVI	**(20R01A04K8)**

Under the esteemed guidance of

Dr. K. NIRANJAN REDDY
(Head of the Department ECE)

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Approved by AICTE, Permanent Affiliation to JNTUH, Accredited by NBA and NAAC
Kandlakoya(V), Medchal Dist - 501 401

www.cmrithyderabad.edu.in

2023-24

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Approved by AICTE, Permanent Affiliation to JNTUH, Accredited by NBA and NAAC
Kandlakoya(V), Medchal Dist - 501 401

www.cmrithyderabad.edu.in

CERTIFICATE

This is to certify that an Industry oriented Mini Project entitled with "DEEP REPRESENTATIONS OF FACE AND FINGERPRINT SPOOFING" is being submitted by:

D.SAKETH	**(20R01A04K5)**
G.VINAY	**(20R01A04K9)**
G.VAISHNAVI	**(20R01A04K8)**

To JNTUH, Hyderabad, in partial fulfillment of the requirement for award of the degree of B.Tech in ECE and is a record of a bonafide work carried out under our guidance and supervision. The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted to have any other University for award of any other degree or diploma.

Project Supervisor	**Project Coordinator**	**HOD , ECE**
Dr. K. Niranjan Reddy	**Mr. G. Venkatswamy**	**Dr. K. Niranjan Reddy**

EXAMINER

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. M Janga Reddy**, Director, **Dr. B. Satyanarayana**, Principal and **Dr. K. Niranjan Reddy**, Head of Department, Dept of Electronics and Communication Engineering, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to **Dr. K. Niranjan Reddy,** Mini Project Coordinator and internal guide, Dept of Electronics and Communication Engineering, CMR Institute of Technology for constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for every step towards success.

D.SAKETH **(20R01A04K5)**
G.VINAY **(20R01A04K9)**
G.VAISHNAVI **(20R01A04K8)**

ABSTRACT

Biometrics systems have significantly improved person identification and authentication, playing an important role in personal, national, and global security. However, these systems might be deceived (or “spoofed”) and, despite the recent advances in spoofing detection, current solutions often rely on domain knowledge, specific biometric reading systems, and attack types. We assume a very limited knowledge about biometric spoofing at the sensor to derive outstanding spoofing detection systems for face, and fingerprint modalities based on two deep learning approaches. The first approach consists of learning suitable convolutional network architectures for each domain, while the second approach focuses on learning the weights of the network via backpropagation. We consider nine biometric spoofing benchmarks each one containing real and fake samples of a given biometric modality and attack type and learn deep representations for each benchmark by combining and contrasting the two learning approaches.

The representations are learned directly from the data with an optimization procedure that randomly searches for the best convolutional neural network from a family of networks defined in a hyperparameter search space. Atop these representations, we couple linear Support Vector Machine classifiers for the final decision making. Instead of learning thousands and/or millions of network weights/parameters based either on backpropagation-like or unsupervised feature learning techniques, our networks use random filter weights initialized in a convenient way, which allows us to quickly probe thousands of network configurations. Experiments on nine publicly available benchmarks of these three modalities show that our framework achieves either very competitive or state-of-the-art results for all problems and modalities

`INDEX

LIST OF FIGURES

LIST OF SCREENSHOTS

1.INTRODUCTION

1.1 ABOUT PROJECT

Biometrics human characteristics and traits can successfully allow people identification and authentication and have been widely used for access control, surveillance, and also in national and global security systems. In the last few years, due to the recent technological improvements for data acquisition, storage and processing, and also the scientific advances in computer vision, pattern recognition, and machine learning, several biometric modalities have been largely applied to person recognition, ranging from traditional fingerprint to face, to iris, and, more recently, to vein and blood flow. Simultaneously, various spoofing attacks techniques have been created to defeat such biometric systems.

There are several ways to spoof a biometric system . Indeed, previous studies show at least eight different points of attack that can be divided into two main groups: direct and indirect attacks. The former considers the possibility to generate synthetic biometric samples, and is the first vulnerability point of a biometric security system acting at the sensor level. The latter includes all the remaining seven points of attacks and requires different levels of knowledge about the system, e.g., the matching algorithm used, the specific feature extraction procedure, database access for manipulation, and also possible weak links in the communication channels within the system.

Given that the most vulnerable part of a system is its acquisition sensor, attackers have mainly focused on direct spoofing. This is possibly because a number of biometric traits can be easily forged with the use of common apparatus and consumer electronics to imitate real biometric readings (e.g., stampers, printers, displays, audio recorders). In response to that, several biometric spoofing benchmarks have been recently proposed, allowing researchers to make steady progress in the conception of anti-spoofing systems. Three relevant modalities in which spoofing detection has been investigated are iris, face, and fingerprint. Benchmarks across these modalities usually share the common characteristic of being image- or video-based.

The success of an anti-spoofing method is usually connected to the modality for which it was designed. In fact, such systems often rely on expert knowledge to engineer features that are able to capture acquisition telltales left by specific types of attacks. However, the need of custom-tailored solutions for the myriad possible attacks might be a limiting constraint. Small changes in the attack could require the redesign of the entire system. For fingerprints, the most common spoofing method consists of using artificial replicas created in a cooperative way, where a mold of the fingerprint is acquired with the cooperation of a valid user and is used to replicate the user’s fingerprint with different materials, including gelatin, latex, play-doh or silicone.

1.2 SPOOFING TECHNIQUES:

1.2.1 FACE SPOOFING:

Face Spoofing, We can categorize the face anti-spoofing methods into four groups: user behavior modeling, methods relying on extra devices, methods relying on user cooperation and, finally, data-driven characterization methods. In this section, we review data-driven characterization methods proposed in literature, the focus of our work herein. System used LBP operator for capturing printing artifacts and micro-texture patterns added in the fake biometric samples during acquisition.This explored color, texture, and shape of the face region and used them with Partial Least Square (PLS) classifier for deciding whether a biometric sample is fake or not. Both works validated the methods with the Print Attack benchmark .

It also explored image-based attacks and proposed the frequency entropy analysis for spoofing detection.Pioneered research on video-based face spoofing detection. They proposed visual rhythm analysis to capture temporal information on face spoofing attacks.

Finally, proposed a score-level fusion strategy in order to detect various types of attacks. In a followup work,proposed an anti-spoofing solution based on the dynamic texture, a spatio-temporal version of the original LBP. Results showed that LBP-based dynamic texture description has a higher effectiveness than the original LBP.

1.2.2 FINGERPRINT SPOOFING:

We can categorize fingerprint spoofing detection methods roughly into two groups: hardware-based (exploring extra sensors) and software-based solutions (relying only on the information acquired by the standard acquisition sensor of the authentication system).A set of feature for fingerprint liveness detection based on quality measures such as 4 ridge strength or directionality, ridge continuity, ridge clarity, and integrity of the ridge-valley structure. The validation considered the three benchmarks used in LivDet 2009 – Fingerprint competition captured with different optical sensors: Biometrika, CrossMatch, and Identix. Later work explored the method in the presence of gummy fingers.

We explored the Weber Local Image Descriptor (WLD) for liveness detection, well suited to high contrast patterns such as the ridges and valleys of fingerprints images. In addition, WLD is robust to noise and illumination changes, and proposed a liveness detection scheme based on Multi-scale Block Local Ternary Patterns (MBLTP). Differently from the LBP, the Local Ternary Pattern operation is done on the average value of the block instead of the pixels being more robust to noise. Binarized Statistical Image Features (BSIF) originally proposed by Kannala. The BSIF was inspired by the LBP and LPQ methods. In contrast to LBP and LPQ approaches, BSIF learns a filter set by using statistics of natural image .Recent results reported in the LivDet 2013 Fingerprint Liveness Detection Competition show that fingerprint spoofing attack detection task is still an open problem with results still far from a perfect classification rate..

1.3 EXISTING SYSTEM:

A facial recognition system is a technology potentially capable of matching a human face from a digital image or a video frame against a database of faces. Such a system is typically employed to authenticate users through ID verification services, and works by pinpointing and measuring facial features from a given image.

Development began on similar systems in the 1960s, beginning as a form of computer application. Since their inception, facial recognition systems have seen wider uses in recent times on smartphones and in other forms of technology, such as robotics. Because computerized facial recognition involves the measurement of a human's physiological characteristics, facial recognition systems are categorized as biometrics. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognition, fingerprint image acquisition, palm recognition or voice recognition, it is widely adopted due to its contactless process.Facial recognition systems have been deployed in advanced human–computer interaction, video surveillance, law enforcement, passenger screening, decisions on employment and housing and automatic indexing of images.

A fingerprint is an impression left by the friction ridges of a human finger. The recovery of partial fingerprints from a crime scene is an important method of forensic science. Moisture and grease on a finger result in fingerprints on surfaces such as glass or metal. Deliberate impressions of entire fingerprints can be obtained by ink or other substances transferred from the peaks of friction ridges on the skin to a smooth surface such as paper. Fingerprint records normally contain impressions from the pad on the last joint of fingers and thumbs, though fingerprint cards also typically record portions of lower joint areas of the fingers.

Human fingerprints are detailed, nearly unique, difficult to alter, and durable over the life of an individual, making them suitable as long-term markers of human identity. They may be employed by police or other authorities to identify individuals who wish to conceal their identity, or to identify people who are incapacitated or deceased and thus unable to identify themselves, as in the aftermath of a natural disaster.

Their use as evidence has been challenged by academics, judges and the media. There are no uniform standards for point-counting methods, and academics have argued that the error rate in matching fingerprints has not been adequately studied and that fingerprint evidence has no secure statistical foundation.

DISADVANTAGES:

- ➢ Issues with recognition of damaged fingerprints.
- ➢ Deployment can be expensive.
- ➢ Fingerprint data can get stolen.
- ➢ Not ideal for remote and field workers.
- ➢ Errors can implicate innocent people.
- ➢ Technology can be manipulated.
- ➢ Misuse causing fraud and other crimes.
- ➢ Threatens privacy.

1.4 PROPOSED SYSTEM:

The proposed system aims to develop a deep representation-based approach for face and fingerprint spoofing detection. This approach leverages the power of deep learning techniques, particularly convolutional neural networks (CNNs), to extract high-level abstract features that effectively differentiate between genuine and spoofed biometric samples. The key components and steps of the proposed system include: Dataset Collection and Preparation: A diverse and comprehensive dataset is collected, consisting of genuine biometric samples and various types of spoofed samples. The dataset includes different materials, lighting conditions, and spoofing techniques to capture real world spoofing scenarios. Deep Representation Learning: CNNs are employed to learn deep representations from the face and fingerprint images. The CNNs are trained using the collected dataset, optimizing their parameters to capture discriminative features that can distinguish between genuine and spoofed samples. Feature Extraction: The trained CNNs are used as feature extractors to obtain deep representations from new input biometric samples.

These deep representations capture the essential characteristics of the samples, allowing for better discrimination between genuine and spoofed data. Classification Model: The extracted deep representations are fed into a classification model, such as support vector machines (SVMs) or deep neural networks (DNNs), to classify the samples as genuine or spoofed. The classification model is trained using the collected dataset, enabling it to learn the distinguishing patterns and make accurate predictions.The system should effectively differentiate between genuine and spoofed samples, accurately detect various types of spoofing attacks, and have the ability to adapt to emerging spoofing techniques. By addressing these challenges, the proposed system aims to enhance the security and reliability of biometric recognition systems in the presence of spoofing attacks.

Several research works have been proposed in the literature to defend from face spoofing attacks. Traditional methods focus on feature descriptors for face spoofing. Deep learning methods focus on high level features. Moreover, it requires a robust model to work efficiently under new imaging and environmental conditions. Existing techniques also use facial feature descriptors for spoofing detection. These features may contain unnecessary information when it is used for face spoofing detection. Moreover, spoofing affects image quality through significant degradation. Considering the limitations in existing techniques, this research work presents an efficient spoofing detection model using texture and quality analysis. The major objective of this research work is to differentiate genuine and spoof images through a classification model, which can be used for many applications that require an efficient authentication process. Entropy and fractal dimension feature descriptors are used to extract the texture and quality features for face spoofing detection.

ADVANTAGES:

- ➢ Fraud prevention.
- ➢ Convenience and efficacy.
- ➢ Improved user experience.
- ➢ Privacy concerns.
- ➢ Enhanced security and safety.
- ➢ A strong line of defense against security breaches.
- ➢ Highly accurate.
- ➢ Unique and can never be same to two persons
- ➢ Authorization or access control

2.REQUIREMENT SPECIFICATIONS:

2.1 REQUIREMENT ANALYSIS

2.1.1 Hardware Requirements

System / PC	Pentium IV 2.4 GHz
Hard Disk	40 GB
Monitor Colour.	15 VGA
Cables	USB to Arduino
EEPROM	512Mb
Arduino	UNO
Fingerprint Module	GTM-5210F52
Camera	OV7670
Display	LCD

2.1.2 Software Requirements:

Operating System	Windows 7
Coding Language	Python, Arduino IDE

2.2 SOFTWARE PRINCIPLES:

2.2.1 Software Specifications

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.
Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the NationalResearch Institute for Mathematics and Computer Science in the Netherlands.
Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.
Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).
Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.Python has a vast and supportive community of developers, data scientists, and machine learning enthusiasts. This community is welcoming to newcomers and provides ample resources for learning. - There are countless tutorials, forums, and documentation available to assist both beginners and experts.Python seamlessly integrates with other data science tools and technologies, such as Jupyter notebooks, which are popular for interactive data analysis and visualization.

Arduino IDE:

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages (Embedded C), using a standard API which is also known as the Arduino Programming Language, inspired by the Processing language and used with a modified version of the Processing IDE. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command line tool developed in Go.

The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

History of Arduino IDE:

The Arduino project was started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy.At that time, the students used a BASIC Stamp microcontroller at a cost of $50. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas. Casey Reas is known for co-creating, with Ben Fry, the Processing development platform. The project goal was to create simple, low cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega128 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller.In 2005, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, extended Wiring by adding support for the cheaper ATmega8 microcontroller. The new project, forked from Wiring, was called Arduino.

2.3 HARDWARE PRINCIPLES:

2.3.1 Components:

Arduino UNO: Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again**.**

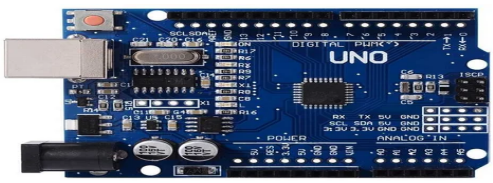

Figure 2.1 Arduino UNO

LCD Module: A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly but instead use a backlight or reflector to produce images in color or monochrome.

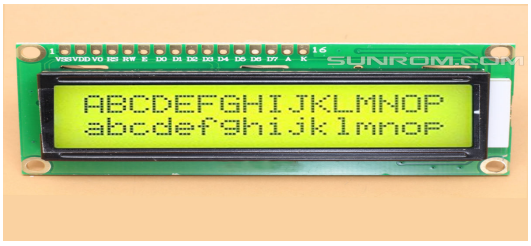

Figure 2.2 LCD Module

Camera Module: The OV7670 camera can be interfaced with an Arduino Uno microcontroller board to capture images and process them in real-time. The OV7670 camera module communicates with the Arduino Uno board via a parallel interface that requires eight data lines and several control signals.

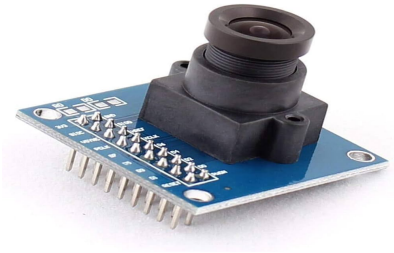

Figure 2.3 Camera Module

Fingerprint Module: Fingerprint sensor module is used as a way to verify identity. It is widely applied to computers, mobile phones, electronic door locks, access control systems, security safes, etc. This ultra-thin optical fingerprint sensor module is designed and manufactured by Jingy technology. It can store and identify up to 3000 fingerprints.

Figure 2.4 Fingerprint Module

Cables: Use it to connect Arduino Uno, Arduino Mega 2560, Arduino 101 or any board with the USB female A port of your computer. Cable length is approximately 100 cm. Cable color and shape may vary slightly from image as our stock rotates**.**

Figure 2.5 USB to Arduino Cable

3.SYSTEM DESIGN

3.1 System Architecture:

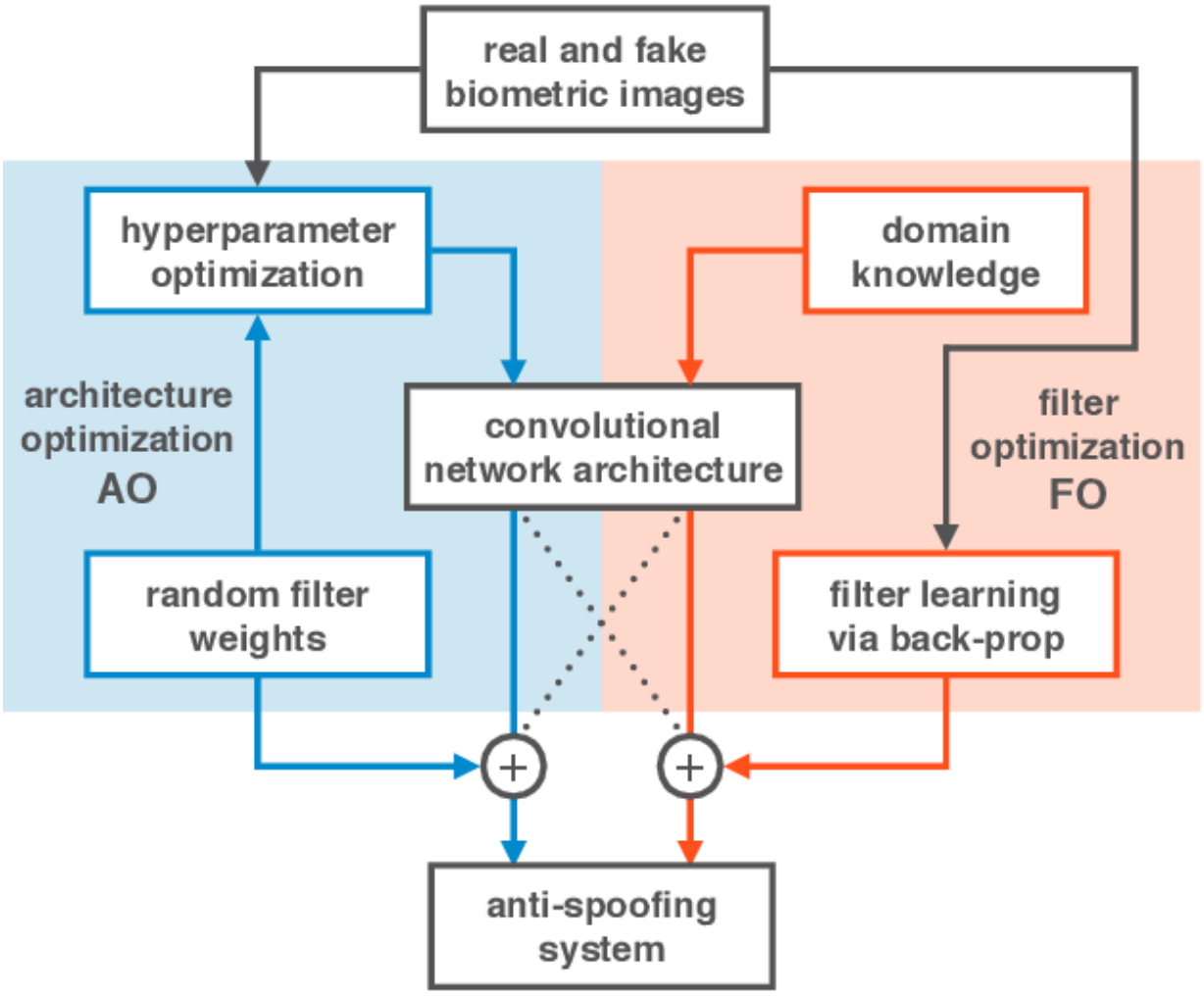

Figure. 3.1. Schematic diagram detailing how anti-spoofing systems are built from spoofing detection benchmarks. Architecture optimization (AO) is shown on the left and filter optimization (FO) on the right. In this work, we not only evaluate AO and FO in separate, but also in combination, as indicated by the crossing dotted lines.

3.2 Block Diagram

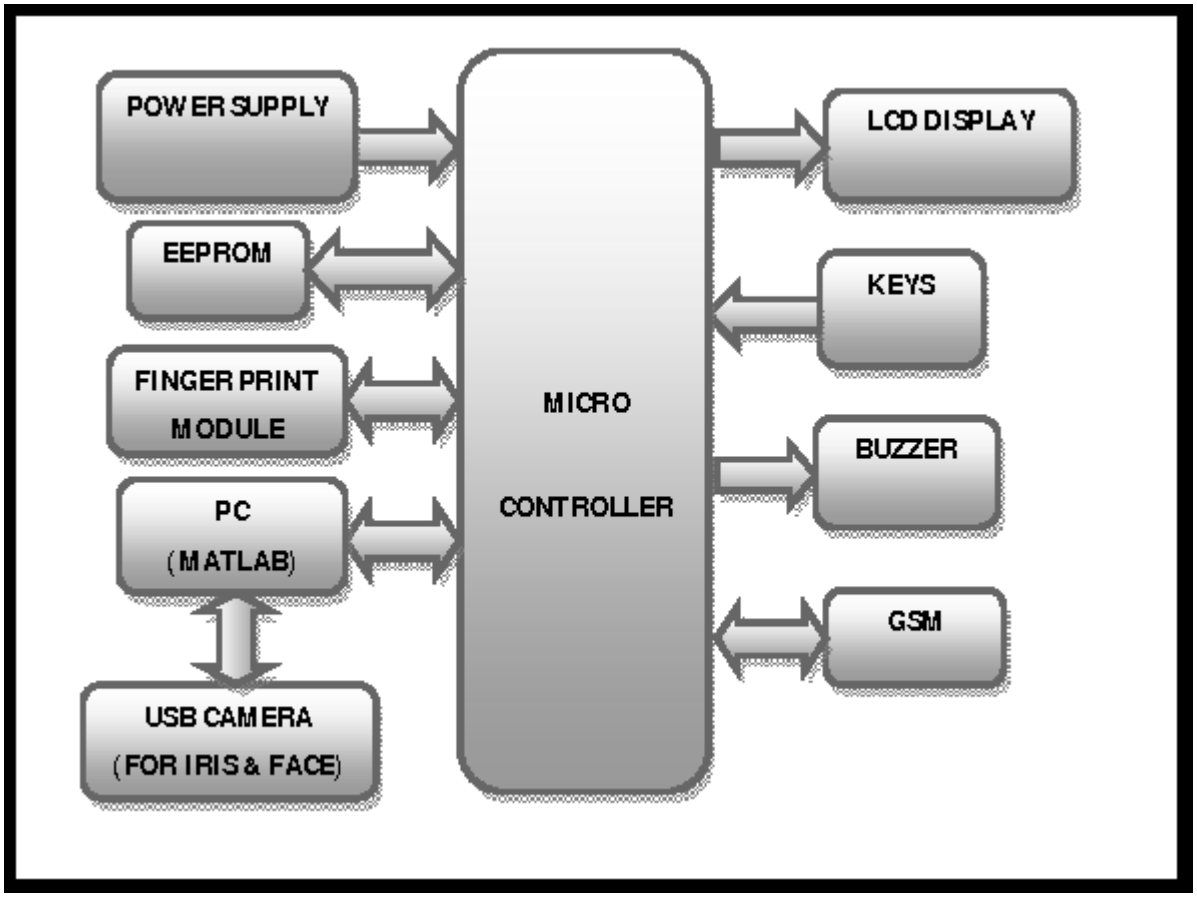

Figure 3.2 Block Diagram

Architecture Optimization (AO): Our approach for AO builds upon the work fundamental, feedforward convolutional operations are stacked by means of hyperparameter optimization, leading to effective yet simple convolutional networks that do not require expensive filter optimization and from which prediction is done by linear support vector machines (SVMs).
Operations in convolutional networks can be viewed as linear and non-linear transformations that, when stacked, extract high level representations of the input. Here we use a well known set of operations called (i) convolution with a bank of filters, (ii) rectified linear activation, (iii) spatial pooling, and (iv) local normalization.

Filter Optimization (FO): We now turn our attention to a different approach for tackling the problem. Instead of optimizing the architecture, we explore the filter weights and how to learn them for better characterizing real and fake samples. Our approach for FO is at the origins of convolutional networks and consists of learning filter weights via the well-known back-propagation algorithm.

For architecture optimization (AO), the decision to use image color information was made according to 10-fold validation, while for filter optimization (FO), color information was considered whenever available for a better approximation with the standard cf10-11 architecture. Finally, images were resized to 32 × 32 or 128 × 128 to be taken as input for the cf10-11 and spoofnet architectures, respectively.

When combined with AO, FO again exceeds previous SOTA in all fingerprint benchmarks and performs remarkably well in MobBIOFake. However, the same difficulty found by FO in previous experiments for both face and two iris benchmarks is also observed here. Even though spoofnet performs slightly better than AO in the cases where SOTA is exceeded, it is important to remark that our AO approach may result in architectures with a much larger number of filter weights to be optimized, and this may have benefited spoof net.

4. IMPLEMENTATION

Our implementation for architecture optimization (AO) is based on Hyperopt-convnet [84] which in turn is based on Theano. LibSVM is used for learning the linear classifiers via Scikit-learn.5 The code for feature extraction runs on GPUs due to Theano and the remaining part is multithreaded and runs on CPUs. We extended Hyperopt-convnet in order to consider the operations and hyperparameters as described. Running times are reported with this software stack and are computed in an Intel i7 @3.5GHz with a Tesla K40 that, on average, takes less than one day to optimize an architecture — i.e., to probe 2,000 candidate architectures.

As for filter optimization (FO), Cuda-convnet is used. This library has an extremely efficient implementation to train convolutional networks via back-propagation on NVIDIA GPUs. Moreover, it provides us with the cf10-11 convolutional architecture taken in this work as reference for FO.

4.1 Texture Analysis:

The widespread use of mobile phones and cameras urges the requirement for incorporating biometric-based authentication systems. Biometrics offers a promising and secure solution for deploying authentication systems. Biometrics is the process of automatic identification of an individual by using behavioral and physical features. Traditional systems utilize the username and password method to login, but the password can be easily shared, lost, forgotten, or hacked. Biometrics offers a large number of applications that range from access control smartcards, criminal investigation, etc. The biometric authentication systems utilize fingerprint, face, and iris features to identify the user. Face is one of the commonly used biometric features due to its simpler and non-intrusive characteristics. A face recognition system is mainly used to identify the user with their face. Face recognition system measures and matches the unique characteristics of the human face for performing both identification and authentication processes. Face recognition systems can be easily compromised and are more vulnerable to spoofing attacks. Spoofing occurs when an unauthorized user attempts to gain access from the face recognition system by using a video or photo of the user.

Spoofing attacks are launched easily by using photo attacks, video replay, and using 3D face masks for different types of attacks. Popularity of social networks makes the photos and images easily accessible. Recent technologies such as plastic surgery and 3D masks are used to spoof and allow the attackers to gain unauthorized access. A printed photo attack occurs when the attacker spoofs the system login with a printed photo of the registered user or genuine user. In few cases, the printed photo is used as a mask such that the high-resolution characterized eyes and mouth are morphed. The imposter will reproduce the eye blinking to create a real facial expression.

Digital photo attack is performed by displaying the photo of a registered/authorized person through a mobile phone or tablet via the front camera. Compared to conventional photo-based attacks, the impact of digital photo-based attacks is less due to device display properties and hence can be identified easily. The video replay attack requires the video from authentic users. Attackers present this video using a digital camera, tablet, and mobile and later play it during the phase of recognition of facial features to gain access to the intended biometric system. The impact of video replay attacks is quite less when compared to print attacks. 3D spoofing attacks are performed by spoofing the system with the help of a mask or clay face. The mask resembles the 3D face shape that features the genuine user. 3D mask attacks are the most advanced attack types and are difficult to identify as compared to other types of attacks. The impact of this type of attack is high and subsequently leads the system into vulnerable conditions.

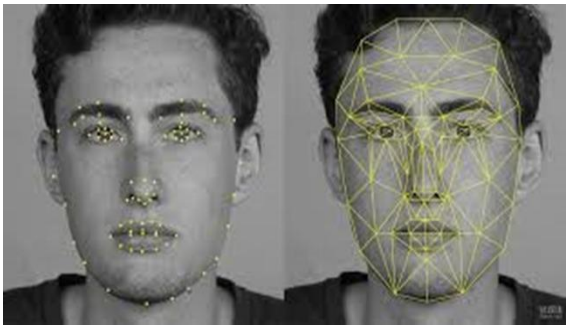

Figure 4.1 Texture Analysis in face

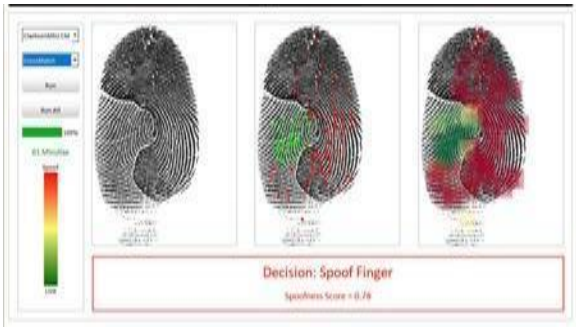

Figure 4.2 Texture Analysis in fingerprint

4.2 Motion Analysis:

Spoofing stratagems keep evolving and present new treats for in-place anti-spoofing countermeasures. The first step toward designing a fake face is to obtain the face print of a target user. This information takes several forms and conditions the type of fake face that can be made. Fake faces are classified into three categories: photos, videos and masks. The easiest way to perform an attack is to steal a photo or a video of a valid client with or without his/her consent. It can be done directly by filming discretely the target client or indirectly via social networks or any on-line media sharing services in which the client is registered. This data can then be utilized to manufacture a fake face. When the user's cooperation is possible, high-quality acquisitions are obtained to manufacture a realistic fake face. In particular, realistic masks can only be achieved by obtaining a high quality 3D scan of the client face so the client's cooperation is required. From a given type of fake face, multiple attack scenarios are possible. When the user's cooperation is possible, high-quality acquisitions are obtained to manufacture a realistic fake face. In particular, realistic masks can only be achieved by obtaining a high quality 3D scan of the client face so the client's cooperation is required. From a given type of fake face, multiple attack scenarios are possible.

When the user's cooperation is possible, high-quality acquisitions are obtained to manufacture a realistic fake face. In particular, realistic masks can only be achieved by obtaining a high quality 3D scan of the client face so the client's cooperation is required. From a given type of fake face, multiple attack scenarios are possible. In the case of mask attacks, if real size masks are employed then the impostor just needs to wear the mask and act normally in front of the sensor following the authentication procedure. In the case of print attacks, the impostor must hold the picture in front of the sensor and may try to simulate real face behavior by manipulating the picture (stretching, bending, moving, …). By cutting some face parts such as eyes and mouth, it is possible to simulate liveness. For video attacks, ideally the screen is fixed so only the replayed motion is visible. The displaying manner is as important as the quality of the fake face when mimicking a real face realistically.

Facing this evolution of spoofing stratagems, numerous anti-spoofing solutions have been developed over the years. Hardware-based protection measures use extra hardware for the anti-spoofing task to measure intrinsic body attributes of the client such as body temperature, heart rate and skin reflectance or body reactions provoked by external stimuli such as eye-blinking in response to a flashing light.

Figure 4.3 Motion Analysis in Face

Figure 4.4 Motion Analysis in Fingerprint

Reversely, software-based countermeasures only use the data acquired by the authentication sensor, usually a webcam, to detect spoofing attacks. Image processing techniques extract the discriminant information between real and fake faces contained in texture, motion or scenic cues.Despite the early popularity of motion-based methods in photo attack detection, motion-based countermeasures have been somehow overlooked due to their lack of robustness against video or mask attacks. Instead, static-based countermeasures have been widely investigated using texture descriptors, band-pass filters and image quality assessment.

4.3 CNN Learning:

Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers are the key component of a CNN, where filters are applied to the input image to extract features such as edges, textures, and shapes. The output of the convolutional layers is then passed through pooling layers, which are used to down-sample the feature maps, reducing the spatial dimensions while retaining the most important information. The output of the pooling layers is then passed through one or more fully connected layers, which are used to make a prediction or classify the image.

CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are not invariant to translation, due to the downsampling operation they apply to the input. Feed-forward neural networks are usually fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) Robust datasets also increase the probability that CNNs will learn the generalized principles that characterize a given dataset rather than the biases of a poorly-populated set.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns to optimize the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered. This independence from prior knowledge and human intervention in feature extraction is a major advantage. A convolutional neural network consists of an input layer, hidden layers and an output layer. In a convolutional neural network, the hidden layers include one or more layers that perform convolutions. Typically this includes a layer that performs a dot product of the convolution kernel with the layer's input matrix. This product is usually the Frobenius inner product, and its activation function is commonly ReLU. As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers.

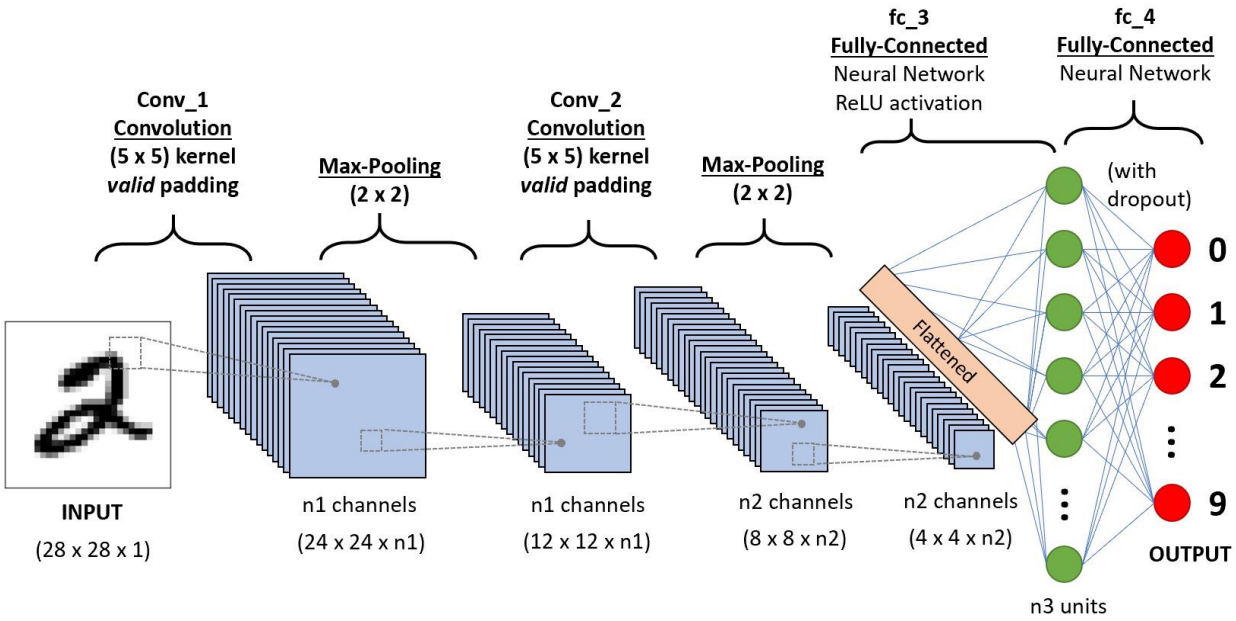

Figure 4.5 CNN Learning

Convolutional neural networks were presented at the Neural Information Processing Workshop in 1987, automatically analyzing time-varying signals by replacing learned multiplication with convolution in time, and demonstrated for speech recognition.

4.4 Source Code:

4.4.1 ARDUINO CODE:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
int fingerprintID = 0;
String IDname;
```

```
void setup(){
    //Fingerprint sensor module setup
    Serial.begin(9600);
    // set the data rate for the sensor serial port
    finger.begin(57600);

    if (finger.verifyPassword()) {
      Serial.println("Found fingerprint sensor!");
    }
    else {
      Serial.println("Did not find fingerprint sensor
  :(");
      while (1) { delay(1); }
    }
  //OLED display setup
    Wire.begin();
    display.begin(SSD1306_SWITCHCAPVCC,
  0x3C);
    //displays main screen
    displayMainScreen();
  }

  void loop(){
    displayMainScreen();
    fingerprintID = getFingerprintIDez();
    delay(50);
    if(fingerprintID == 1 || fingerprintID == 3 ||
  fingerprintID == 4 || fingerprintID == 5){
      IDname = "Sara";
      displayUserGreeting(IDname);
    }
    else if(fingerprintID == 2){
      IDname = "Rui";
      displayUserGreeting(IDname);
    }
  }
  / / returns -1 if failed, otherwise returns ID #
  int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK)  return -1;
   p = finger.image2Tz();
    if (p != FINGERPRINT_OK)  return -1;


p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;
```

```
  // found a match!
  Serial.print("Found ID #");
  Serial.print(finger.fingerID);
  Serial.print(" with confidence of ");
  Serial.println(finger.confidence);
  return finger.fingerID;
}
void displayMainScreen(){
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(7,5);
  display.println("Waiting fingerprint");
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(52,20);
  display.println("...");
  display.display();
  delay(2000);
}
void displayUserGreeting(String Name){
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(2);
  display.setCursor(0,0);
  display.print("Hello");
  display.setCursor(0,15);
  display.print(Name);
  display.display();
  delay(5000);
  fingerprintID = 0;
}
```

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306
display(OLED_RESET);
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3)
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
int fingerprintID = 0;
String IDname;
void setup(){
 //Fingerprint sensor module setup
 Serial.begin(9600);
 // set the data rate for the sensor serial port
 finger.begin(57600);

 if (finger.verifyPassword()) {
   Serial.println("Found fingerprint sensor!");
 }
else {
   Serial.println("Did not find fingerprint sensor :(");
   while (1) { delay(1); }
 }
//OLED display setup
 Wire.begin();
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
 //displays main screen
 displayMainScreen();
}
if(fingerprintID == 1 || fingerprintID == 3 ||
fingerprintID == 4 || fingerprintID == 5){
 IDname = "Sara";
```

`

```
displayUserGreeting(IDname);
}
else if(fingerprintID == 2){
  IDname = "Rui";
void setup() {
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4,OUTPUT);
Serial.begin(9600); // start serial communication at
9600bps
bluetooth.begin(9600);
}

void loop() {
Serial.println(value);
if (bluetooth.available())
{
value = bluetooth.readString();

if (value == “all led turn on”){
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
}
if (value == “all led turn off”){
digitalWrite(2, LOW);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
}

if (value == “turn on red led”){
digitalWrite(2, HIGH);
}
if (value == “turn on the fan”){
digitalWrite(3, HIGH);
}
```

```
if (value == “turn off red led”){
digitalWrite(2, LOW);
}
if (value == “turn on blue led”){
digitalWrite(4, HIGH);
}
if (value == “turn off red led”){
digitalWrite(4, LOW);
}
if (value == “turn off the fan”){
digitalWrite(3, LOW);
}
}
}
```

4.4.2 PYTHON CODE:

```
pip install dlib
# installing face recognition
pip install face recognition
import cv2
import numpy as np
import face_recognition
import cv2
import numpy as np
import face_recognition
img_bgr =
face_recognition.load_image_file('student
_images/modi.jpg')
img_rgb =
cv2.cvtColor(img_bgr,cv2.COLOR_BGR
2RGB)
cv2.imshow('bgr', img_bgr)
cv2.imshow('rgb', img_rgb)
cv2.waitKey
img_modi=face_recognition.load_image_file('student_images/modi.jpg')
img_modi_rgb = cv2.cvtColor(img_modi,cv2.COLOR_BGR2RGB)
```

```
history = History()
df=pd.read_csv('data1modify.csv')
df=df.dropna()
a=list(df.columns)
df1=df[[a[0],a[1],a[2],a[3],a[4],a[5]]]
Y = df[a[6]]
print(df)

model = Sequential()
model.add(Dense(20, activation='sigmoid', input_shape=(6,)))
model.add(Dense(10, activation='sigmoid'))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam',
metrics=['mse'])
print(model.summary())

X_train, X_test, Y_train, Y_test = train_test_split(df1,Y, test_size=0.2,
random_state=42)
history1 = model.fit(X_train,
Y_train,validation_data=(X_test,Y_test),epochs=100,batch_size=100,verbos
e=2, callbacks = [history])
print(X_test)
print(Y_test)
predictions = model.predict(X_test)
print(predictions)
model.save('model_trial.h5')
from keras.models import Sequential
from keras.layers import Dense,Input
from keras.callbacks import History
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn import metrics
cover()': x4, 'wind speed(mph)': x5, 'wind dir.':x6}]
X_test = pd.DataFrame(input_value)
predictions = loaded_model.predict(X_test)
if(predictions[0]<0):
predictions[0]=0
model.save('model_trial.h5')
```

5. TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement

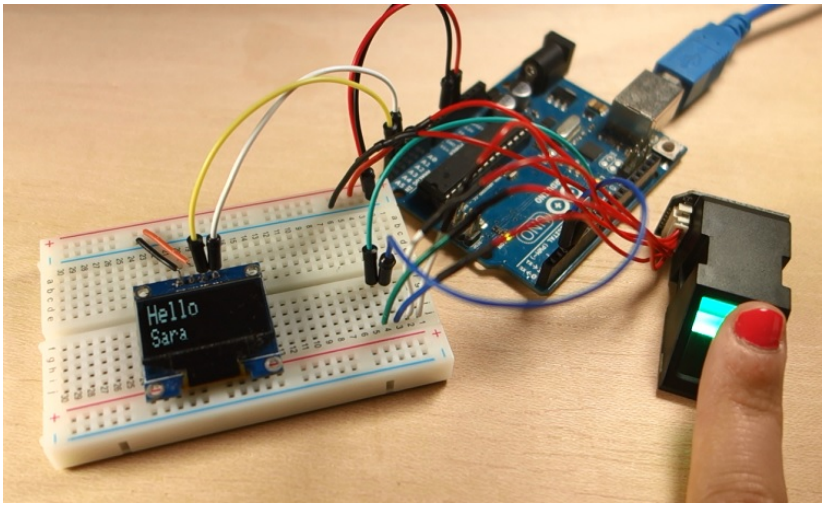

Figure 5.1 Fingerprint Detection

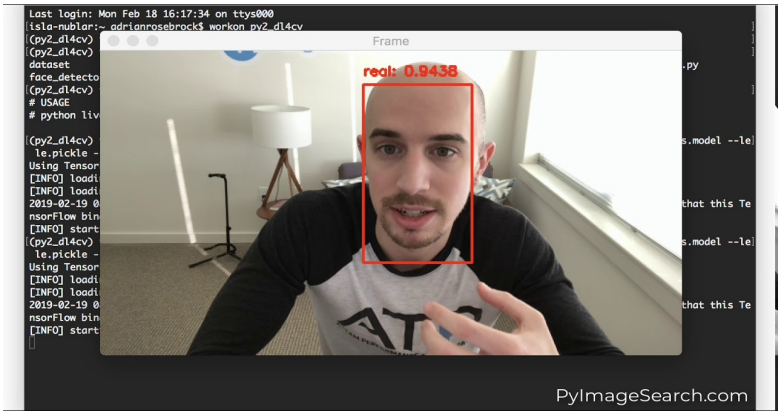

Figure 5.2 Face Detection

.

6.RESULT

We evaluate the effectiveness of the proposed methods for spoofing detection. We show experiments for the architecture optimization and filter learning approaches along with their combination for detecting iris, face, and fingerprint spoofing on the nine benchmarks described. We also present results for the *spoof net*, which incorporates some domain-knowledge on the problem. We compare all of the results with the state-of-the-art counterparts. Finally, we discuss the pros and cons of using such approaches and their combination along with efforts to understand the type of features learned and some efficiency questions when testing the proposed methods

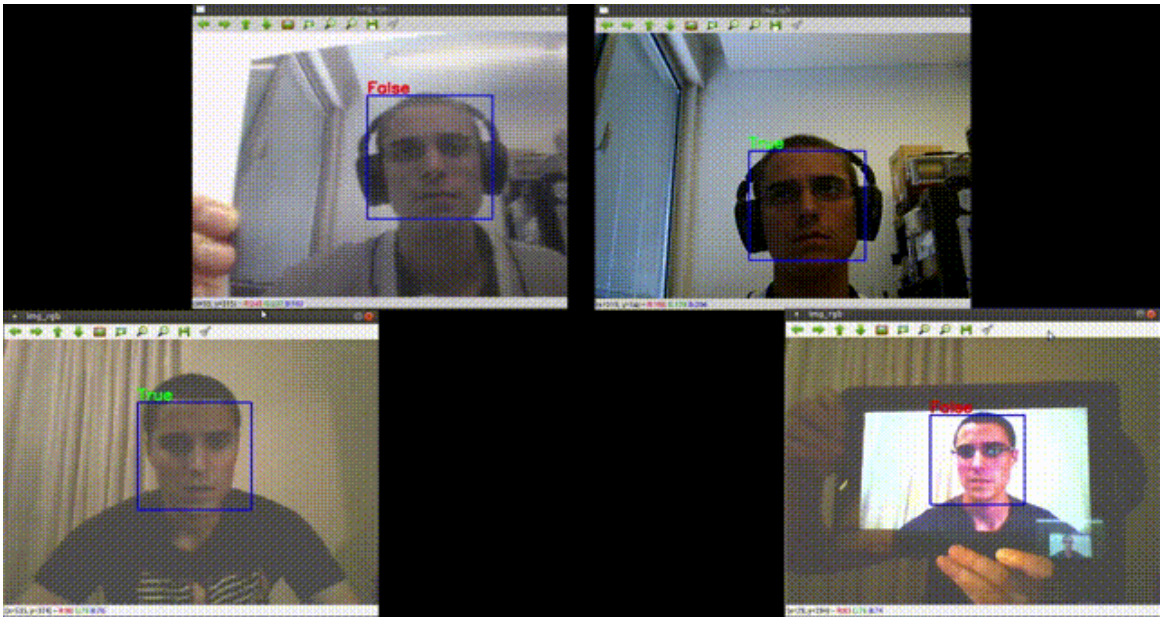

Figure 5.3 Results of Face Detection

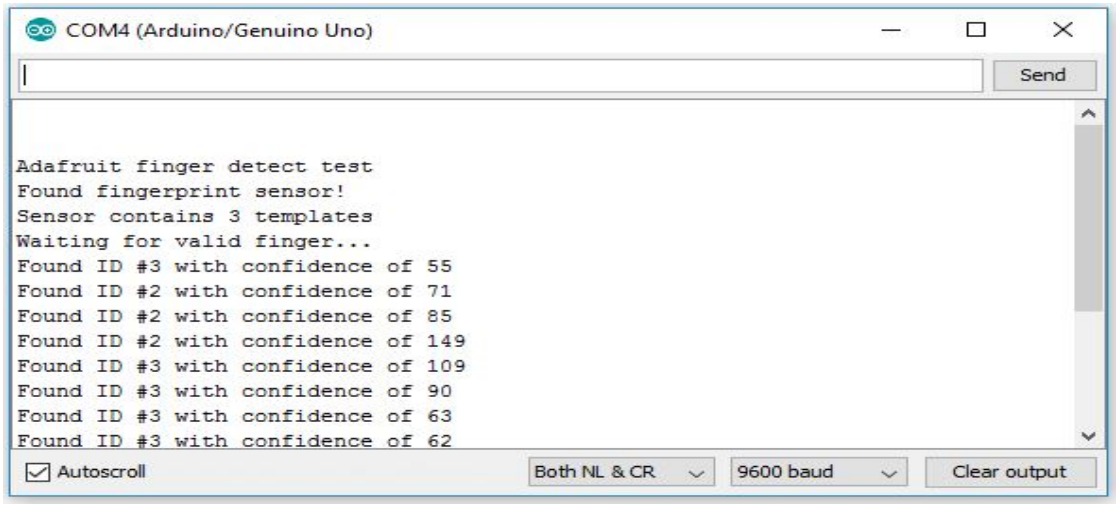

Figure 5.4 Results of Fingerprint Detection

7.CONCLUSION

In this work, we investigated two deep representation re- search approaches for detecting spoofing in different biometric modalities. On one hand, we approached the problem by learning representations directly from the data through architecture optimization with a final decision-making step atop the representations. On the other, we sought to learn filter weights for a given architecture using the well-known back- propagation algorithm. As the two approaches might seem naturally connected, we also examined their interplay when taken together. In addition, we incorporated our experience with architecture optimization as well as with training filter weight for a given architecture into a more interesting and adapted network, spoof net.

As the data tell it all, the decision to which path to follow can also come from the data. Using the evaluation/validation set during training, the researcher/developer can opt for optimizing architectures, learning filters or both. If training time is an issue and a solution must be presented overnight, it might be interesting to consider an already learned network that incorporates some additional knowledge in its design. In this sense, a spoof net could be a good choice. In all cases, if the developer can incorporate more training examples, the approaches might benefit from such augmented training data. The proposed approaches can also be adapted to other biometric modalities not directly dealt with herein. The most important difference would be in the input type of data since all discussed solutions directly learn their representations from the data.

For the case of iris spoofing detection, here we dealt only with iris spoofing printed attacks and some experimental datasets using cosmetic contact lenses have recently become available allowing researchers to study this specific type of spoofing. For future work, we intend to evaluate such datasets using the proposed approaches here and also consider other biometric modalities such as palm, vein, and gait. It is important to emphasize the interplay between the architecture and filter optimization approaches for the spoofing problem. It is well-known in the deep learning literature that when thousands of samples are available for learning, the filter learning approach is a promising path. Indeed, we could corroborate this through fingerprint benchmarks that consider a few thousand samples for training. However, it was not the case for faces and two iris benchmarks which suffer from the small sample size problem (SSS) and subject variability hindering the filter learning process. In these cases, the architecture optimization approach was able to learn representative and discriminative features providing comparable spoofing effectiveness to the SOTA results in almost all benchmarks, and specially outperforming them in three out of four SOTA results when the filter learning approach failed. It is worth mentioning that sometimes it is still possible to learn meaningful features from the data even with a small sample size for training. We believe this happens in more well-posed datasets with less variability between training/testing data as it is the case in which the AO approach achieved 99.38% just 0.37% behind the SOTA result.

Finally, it is important to take all the results discussed herein with a grain of salt. We are not presenting the final word in spoofing detection. In fact, there are important additional research that could finally take this research another step forward. We envision the application of deep learning representations on top of pre-processed image feature maps (e.g., LBP-like feature maps, acquisition-based maps exploring noise signatures, visual rhythm representations, etc.). With an n-layer feature representation, we might be able to explore features otherwise not possible using the raw data. In addition, exploring temporal coherence and fusion would be also important for video-based attacks.

8. REFERENCES

- A. K. Jain and A. Ross, Handbook of Biometrics. Springer, 2008, Introduction to biometrics, pp. 1–22.
- C. Rathgeb and A. Uhl, "Attacking iris recognition: An efficient hill- climbing technique," in IEEE/IAPR International Conference on Pattern Recognition (ICPR), 2010, pp. 1217–1220.
- ——, "Statistical attack against iris-biometric fuzzy commitment schemes," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011, pp. 23–30.
- J. Galbally, J. Fierrez, and J. Ortega-garcia, "Vulnerabilities in biometric systems: Attacks and recent advances in liveness detection," Database, vol. 1, no. 3, pp. 1–8, 2007, available at http://atvs.ii.uam.es/files/2007 SWB VulnerabilitiesRecentAdvances Galbally.pdf.
- N. K. Ratha, J. H. Connell, and R. M. Bolle, "An analysis of minutiae matching strength," in International Conference on Audio-and Video- Based Biometric Person Authentication, 2001, pp. 223–228.
- W. Robson Schwartz, A. Rocha, and H. Pedrini, "Face spoofing detection through partial least squares and low-level descriptors," in IEEE Int. Joint Conference on Biometrics (IJCB), 2011, pp. 1 –8.
- D. Yi, Z. Lei, Z. Zhang, and S. Li, "Face anti-spoofing: Multi-spectral approach," in Handbook of Biometric Anti-Spoofing, ser. Advances in Computer Vision and Pattern Recognition, S. Marcel, M. S. Nixon, andS. Z. Li, Eds.S. Z. Li, Eds. Springer London, 2014, pp. 83–102.
- J. Ma¨a¨tta¨, A. Hadid, and M. Pietika¨inen, "Face spoofing detection from single images using micro-texture analysis," in IEEE Int. Joint Conference on Biometrics (IJCB), 2011, pp. 1 –7.
- A. K. Jain and A. Ross, Handbook of Biometrics. Springer, 2008, Introduction to biometrics, pp. 1–22.
- C. Rathgeb and A. Uhl, "Attacking iris recognition: An efficient hill- climbing technique," in IEEE/IAPR International Conference on Pattern Recognition (ICPR), 2010, pp. 1217–1220.
- ——, "Statistical attack against iris-biometric fuzzy commitment schemes," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011, pp. 23–30.
- J. Galbally, J. Fierrez, and J. Ortega-garcia, "Vulnerabilities in biometric systems: Attacks and recent advances in liveness detection," Database, vol. 1, no. 3, pp. 1–8, 2007, available at http://atvs.ii.uam.es/files/2007 SWB VulnerabilitiesRecentAdvances Galbally.pdf.
- N. K. Ratha, J. H. Connell, and R. M. Bolle, "An analysis of minutiae matching strength," in International Conference on Audio-and Video- Based Biometric Person Authentication, 2001, pp. 223–228.
- W. Robson Schwartz, A. Rocha, and H. Pedrini, "Face spoofing detection through partial least squares and low-level descriptors," in IEEE Int. Joint Conference on Biometrics (IJCB), 2011, pp. 1 –8.
- D. Yi, Z. Lei, Z. Zhang, and S. Li, "Face anti-spoofing: Multi-spectral approach," in Handbook of Biometric Anti-Spoofing, ser. Advances in Computer Vision and Pattern Recognition, S. Marcel, M. S. Nixon, andS. Z. Li, Eds.S. Z. Li, Eds. Springer London, 2014, pp. 83–102.
- J. Ma¨a¨tta¨, A. Hadid, and M. Pietika¨inen, "Face spoofing detection from single images using micro-texture analysis," in IEEE Int. Joint Conference on Biometrics (IJCB), 2011, pp. 1 –7.
- M. Gu¨nther, D. Haufe, and R. P. Wu¨rtz, "Face recognition with disparity corrected gabor phase differences," in *Int. Conference on Artificial Neural Networks and Machine Learning (ICANN)*, 2012, pp. 411–418.
- N. Erdogmus and S. Marcel, "Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect," in *IEEE Int. Conference on Biometrics: Theory Applications and Systems (VISAPP)*, 2013, pp. 1–6.
- N. Kose and J.-L. Dugelay, "On the vulnerability of face recognition systems to spoofing mask attacks," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 2357–2361
- ——, "Countermeasure for the protection of face recognition systems against mask attacks," in *IEEE Int. Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2013, pp. 1–6.
- X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 504–517.
- N. Kose and J.-L. Dugelay, "Reflectance analysis based countermeasure technique to detect face mask attacks," in *Int. Conference on Digital Signal Processing (DSP)*, 2013, pp. 1–6.
- T. de Freitas Pereira, A. Anjos, J. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?" in *IAPR Int. Conference on Biometrics (ICB)*, 2013, pp. 1–8.
- T. Freitas Pereira, J. Komulainen, A. Anjos, J. De Martino, A. Hadid,M. Pietikainen, and S. Marcel, "Face liveness detection using dynamic texture," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 2, 2014.
- L. Ghiani, D. Yambay, V. Mura, S. Tocco, G. Marcialis, F. Roli, and S. Schuckcrs, "Livdet 2013 – fingerprint liveness detection competition," in *International Conference on Biometrics (ICB)*, 2013, pp. 1–6. [Online]. Available: http://prag.diee.unica.it/fldc/
- G. L. Marcialis, A. Lewicke, B. Tan, P. Coli, D. Grimberg,A. Congiu, A. Tidu, F. Roli, and S. A. C. Schuckers, "Livdet 2009– first international fingerprint liveness detection competition," in Int. Conference on Image Analysis and Processing (ICIAP), ser. Lecture Notes in Computer Science, P. Foggia, C. Sansone, and M. Vento, Eds., vol. 5716. Springer, 2009, pp. 12–23. [Online]. Available: http://prag.diee.unica.it/LivDet09
- J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia, "A high performance fingerprint liveness detection method based on quality related features," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 311–321, 2012.
- D. Yambay, L. Ghiani, P. Denti, G. Marcialis, F. Roli, and S. Schuckers, "Livdet 2011 – fingerprint liveness detection competition," in *IAPR Int. Conference on Biometrics (ICB)*, 2012, pp. 208–215. [Online]. Avail- able: http://people.clarkson.edu/projects/biosal/fingerprint/index.php