UNIVERSIDADE D
COIMBRA

André Pereira Graça

Liveness Detection and Facial Recognition with Multi-Modal Features

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, ramo de Automação, orientada pelo Professor Doutor Jorge Manuel Moreira de Campos Pereira Batista e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Coimbra, Outubro de 2021

UNIVERSIDADE D
COIMBRA

LIVENESS DETECTION AND FACIAL RECOGNITION WITH MULTI-MODAL FEATURES

André Pereira Graça

Coimbra, October of 2021

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

LIVENESS DETECTION AND FACIAL RECOGNITION WITH MULTI-MODAL FEATURES

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, ramo de Automação, orientada pelo Professor Doutor Jorge Manuel Moreira de Campos Pereira Batista e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Supervisor:

Prof. Dr. Jorge Manuel Moreira de Campos Pereira Batista

Jury:

Prof. Dr. Jorge Manuel Moreira de Campos Pereira Batista

Prof. Dr. Luís Alberto da Silva Cruz

Prof. Dr. Nuno Miguel Mendonça da Silva Gonçalves

Coimbra, October of 2021

Agradecimentos

O trabalho que aqui é apresentado não seria possível sem o apoio de todos aqueles com quem convivo diariamente.

Gostaria de começar por agradecer ao meu orientador toda a sua ajuda, as discussões enriquecedoras e os conhecimentos que foram fundamentais para a concretização deste trabalho.

À equipa do laboratório, em especial ao Bruno Silva e ao Pedro Martins a constante disponibilidade e ajuda. Um especial agradecimento a todos os participantes nos testes da minha tese pela boa vontade sempre que era pedido.

A toda a minha família em especial à minha Mãe, ao meu Pai, aos meus Irmãos e aos meus Avós por todo o apoio e força ao longo de todo o meu percurso.

À Sara, um grande obrigado pelo apoio e motivação e por estares sempre presente quando era preciso.

A todos os meus amigos, obrigado pela amizade, pelo apoio e entreajuda fundamental para chegar até ao fim deste curso. Um grande obrigado por todos os momentos vividos ao longo dos últimos 5 anos.

A todos aqueles que fizeram parte deste meu percurso e não foram mencionados até aqui! À cidade que é Coimbra!

A todos, um muito obrigado!

Abstract

The interest in the topic of Liveness Detection has been increasing in the past few years due to the development of new tools and knowledge in the area of biometry. Liveness Detection is essential in user authentication systems to stop intruders from gaining access to confidential information illegally. The face authentication systems we have today can be victims to a simple photograph of a legitimate user, which can be easily obtained through social media networks, video replay, or 3D masks.

Various face anti-spoofing algorithms have been proposed to tackle this problem, using different approaches, and numerous public face anti-spoofing databases and competitions.

Face recognition accuracy is significantly improved using deep learning networks due to their ability to extract human faces deep features.

The combination of liveness features from the image visual cues provides a better generalization for a face anti-spoofing classifier, taking advantage of the feature fusion or score fusion approach, used by all the state-of-the-art face anti-spoofing measures.

This dissertation aimed to create a Liveness Detection and Facial Recognition system based on convolutional neural networks (CNN) and using information collected from 3 different modalities (RGB, Infra-Red, and Depth images). The use of multiple modalities in this context is still being explored, and there is still much development needed to achieve the perfect biometric authentication system.

To evaluate the accuracy of our system, we performed evaluations on the CASIA-SURF Dataset, which is a well-known multi-modality dataset, and we also created a dataset precisely for this task. We tested our network with different network architectures and modules and implemented a biometric system that can work in the real world in real-time.

The results were promising, showing the possibility of using this system for user authentication in the real world.

KEYWORDS: Liveness Detection, Multi-modalities, Facial Recognition, Real-Time, Biometrics

Resumo

O interesse pelo tema da detecção de vivacidade tem vindo a aumentar nos últimos anos devido ao desenvolvimento de novas ferramentas e conhecimentos na área da biometria. A detecção de vivacidade é essencial em sistemas de autenticação de identidade para impedir que alguém tenha acesso a informações confidenciais ilegalmente. Os sistemas de autenticação facial existentes podem ser enganados através de uma simples fotografia de um usuário legítimo, que pode ser facilmente obtida por meio de redes sociais, vídeo ou máscaras 3D. Vários algoritmos com o objetivo de deteção de vivacidade têm sido propostos para lidar com este problema, usando diferentes abordagens e numerosas bases de dados.

A precisão do reconhecimento facial é significativamente melhorada usando redes neuronais devido à sua capacidade de extrair características do rosto visivelmente imperceptíveis. A combinação da fusão de caracteristicas e abordagens de fusão de modalidades fornece uma melhor generalização para sistemas de deteção de vivacidade, qualidade esta que qualquer método de deteção de vivacidade facial mais recente ambiciona.

Esta dissertação teve como objetivo a criação de um sistema de detecção de vivacidade e reconhecimento facial baseado em redes neuronais convolucionais (RNC) e utilizando informação obtida por 3 diferentes modalidades (imagens de cor, infra-vermelhos e profundidade). O uso de múltiplas modalidades neste contexto ainda está a ser explorado, e ainda há muito desenvolvimento necessário para alcançar um perfeito sistema de autenticação biométrica.

Para avaliar a precisão do nosso sistema, utilizamos a base de dados CASIA-SURF com imagens faciais provenientes de diferentes modalidades, e também criamos uma base de dados especificamente para esta tarefa. Testamos a nossa rede com diferentes arquiteturas e vários módulos e implementamos um sistema biométrico que pode funcionar em situações fora do laboratório em tempo real.

Os resultados foram promissores, mostrando a possibilidade da utilização desse sistema para autenticação de usuários no mundo real.

PALAVRAS-CHAVE: Deteção de vivacidade, Multiplas-modalidades, Reconhecimento Facial, Tempo-Real, Biometria

Contents

List of Tables

List of Figures

List of Abbreviations

SA - Spoofing Attack

FAS - Face Anti-Spoofing

CNN - Convolutional Neural Network

NN - Neural Network

PAD - Presentation Attack Detection

ANN - Artificial Neural Network

NSL - Normalized Softmax Loss

LMCL - Large Margin Cosine Loss

ReLU - Rectified Linear Unit

FC - Fully Connected

GAP - Global Average Pooling

SE - Squeeze and Excitation

GUI – Graphic User Interface

FIFO – First In First Out

ROC - Receiver Operating Characteristic

FR – Facial Recognition

LD – Liveness Detection

1

Introduction

Face recognition has been widely applied in user authentication systems, among various biometric techniques, such as fingerprinting, iris scanning, and hand geometry, due to the non-intrusive, natural face biometrics interaction and low cost.

Many of the latest facial recognition technologies are vulnerable to spoofing attacks (SA), which occur when someone tries to bypass a face biometric system by presenting a false face in front of the camera.

In the past years, the tech world has witnessed significant breakthroughs in deep learning in various real-world applications, such as face recognition, access control, phone unlock, and face payment. With current face authentication systems, a simple photograph of a legitimate user, which can be easily obtained through social media networks, video replay, or 3D mask, can spoof a face recognition system to access confidential information illegally. Consequently, the ability to separate the real faces from the spoofs is urgently needed to reduce security concerns. To tackle this problem, a variety of face anti-spoofing algorithms were proposed based on different approaches, and numerous public face anti-spoofing databases and competitions were created, promoting the development in face liveness detection.

The techniques used can generally be divided into two categories according to the face feature extracting methodology: methods that manually extract features based on traditional machine learning and those that automatically acquire face features based on deep learning.

The accuracy of face recognition is significantly improved using deep learning networks due to their ability to extract human faces deep features.

The combination of liveness features from the image visual cues provides a better generalization for a face anti-spoofing classifier, taking advantage of the feature fusion or score fusion approach, used by all the state-of-the-art face anti-spoofing measures. These methods suffer from overfitting and poor generalization to new patterns and environments, so multi-modal data analysis has been studied and applied to tackle these difficulties, which further

utilizes the infrared (IR) spectral image or Depth image to improve the face anti-spoofing (FAS) detection.

1.1. Context and Motivation

The modern world brings a new era of security involving Liveness Detection and Facial Recognition. These features are applied to a vast spectrum of applications, mainly to access control, and with these innovations, new ways to spoof these systems are also created [1].
With the support of artificial intelligence, it is possible to create a better system that is not so easily spoofed. It is necessary to make sure it is a real person trying to use the application to correctly use facial recognition. This can be accomplished with different methods, such as asking the user to blink or relying on some algorithm to identify skin textures or cut-outs in masks, or in this case, by using a combination of three different information modalities (RGB, IR, and Depth data). After this condition is met, it is possible to proceed to the facial recognition phase. In this phase, there are also several ways to proceed, such as analyzing a person's facial features, like the shape of the nose or the distance between the eyes or any other anthropometric measurements.

1.2. Goals and Contributions

The major goal of this dissertation is the development of a liveness detection and facial recognition (LDFR) approach using convolutional neural networks (CNN) to distinguish real images from spoofed images as observed in Figure 1. For this solution we used information collected from 3 different modalities (RGB, Near Infra-Red, and Depth images). The data collected from the three modalities differ in importance for each task: liveness detection and facial recognition. It has been proven that Neural Networks (NN) using only one or two of the three modalities are more susceptible to classifying a face wrongly, either causing an increase in false positives or false negatives. RGB data has rich appearance details, but many Presentation Attack Detection (PAD) methods fail when facing new types of attacks, such as 3D and custom-made silicone masks. IR data measures the amount of heat radiated from a face, a feature that is uniquely from a real face. The Depth data is important to differ the real faces, which have some unique cleavage, from for example, a flat printed image of a face. Combining all the information present in all the modalities features allows the neural network to be more reliable and resistant to spoof attacks, providing a better chance of classifying the spoof and real faces correctly so the information's re-weighting will play a significant role in developing this dissertation.

The course flow of this dissertation and content of each chapter are the following:

- Chapter 2 reviews the state of the art of liveness detection and facial recognition systems, the different types of methods and datasets available, as well as the various studies that use biometric identification and authentication with different modalities and different metrics;
- Chapter 3 presents the general concepts about CNNs as well as the facial recognition and liveness detection methods and loss metrics studied in this work;
- Chapter 4 describes the entire architecture of the identification pipeline used for the proposed architecture and the implementation of the system in real-time for access control in the real world, as well as details about the making of our private dataset;
- Chapter 5 describes the validation methods, datasets and metrics used and presents the results obtained for the different architectures used;
- Chapter 6 draws some conclusions and proposes some improvements for future work.

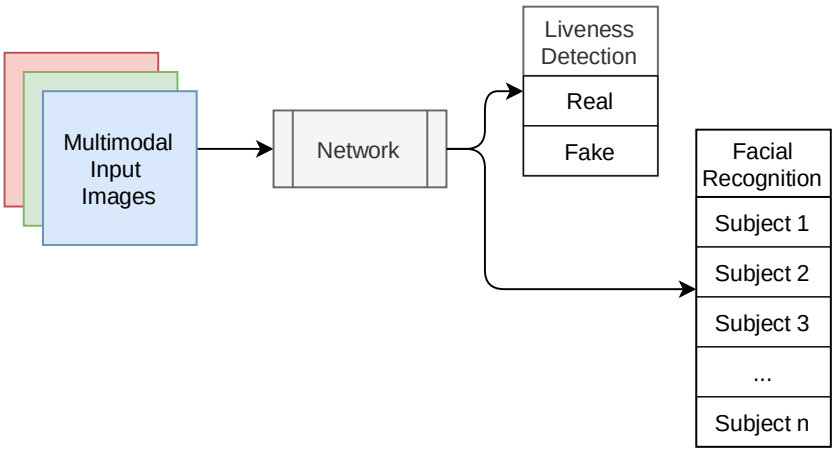

Figure 1 - Proposed pipeline

2

State of The Art

In the last years, the number of publications focused on liveness detection and facial recognition has been increasing exponentially. In this chapter, we present the most relevant applications and methods related to liveness detection and facial recognition.

2.1. Liveness Detection

Due to its adaptability and exceptional accuracy, face recognition technology [2] has been applied in a range of different applications such as border control [1], mobile payment, and checking-in. However, the systems using this type of application are vulnerable to presentation attacks (PA) such as print attacks, video replay attacks, makeup, 3D masks, and deep fakes. According to the Biometrics Glossary[1], liveness detection is a technique that confirms or denies the presence of a real user in a biometric sample, securing any biometric system from different types of spoofing attacks. Spoofing is the ability to mislead a biometric sensor into recognizing an incorrect biometric subject as a valid subject.

Most popular algorithms are created based on human liveness cues and explored for dynamic differentiation, which requires rich task-aware prior knowledge for design, such as face and head movement (e.g., nodding, smiling, eye-blinking) and remote physiological signals (e.g., rPPG - remote photoplethysmography [2]). However these types of liveness cues are usually obtained from interactive face videos, which is inconvenient for practical usage, and also because these cues are easily simulated by video attacks, making them less reliable.

Considering this problem, the appearance of large-scale public Face Anti-Spoofing (FAS) datasets with various attack types and recorded sensor modalities also improve today's anti-spoofing research. New face image datasets with several subjects and samples have been created for this task, such as CelebA-Spoof [3], which contains over 500k images recorded

[1] https://www.hsdl.org/?abstract&did=464490
[2] https://www.noldus.com/blog/what-is-rppg

from over 10k subjects. Besides the traditional PA types, print and replay attacks, some up-to-date datasets, such as SiW-M [4], contain more than 10 PA types, yet with fewer subjects. Regarding sensor modality and other recording hardware, most researches mainly focus on a single RGB modality and overlook the deep learning applications on the multi-modal sensors, which significantly benefit the spoofing detection task. The CASIA-SURF [5] approach to this task confirms this effectiveness by fusing RGB, Depth, and Near-IR data.

2.1.1. Liveness Detection Methods

Previous face anti-spoofing approaches use a binary classifier problem (e.g., '0' for spoofing while '1' for live faces) to identify the real face or spoof, which can be categorized into several categories:

- extra hardware-aided
- motion-based
- image quality-based
- deep-learning-based methods.

2.1.1.1. Extra hardware-aided-based methods

In addition to 2D images captured by regular cameras in the visible spectrum, these methods involve the use of extra hardware that can render other vital information to distinguish between real and fake faces. In Lagorio *et al.*, the authors obtained 3D scanned points of face surfaces and calculated 3D curvatures from the collected data to identify real faces and 2D spoofings [6]. L. Sun *et al.* obtained RGB and the corresponding infrared images from a face simultaneously, with the combination of a regular and a thermal camera to detect face liveness [7].

2.1.1.2. Motion-based methods

These methods aim to classify face videos based on the movements of either facial parts or the scene. Though these methods are effective against photo attacks they become vulnerable when attackers simulate these motions through paper cut-outs. Furthermore, most motion-based methods need to extract optical flow, which is a time-consuming process, limiting the use cases.

Based on the region examined, motion-based methods can be grouped into two sub-categories, face motion-based and scene motion-based.

A real human face shows a different optical flow trajectory in comparison with a 2D photo face. Blinking the eyes is a common face motion cue highly used in early works. Bharadwaj *et al.* used Eulerian motion magnification to intensify precise facial motions [8]. With this achievement, it was possible to exaggerate micro and macro-facial expressions from real and fake faces, and distinguish the real motion pattern from the misrepresented motion patterns, thus identifying spoofs.

The non-existence of motion between the user and background can indicate the presence of a spoofing attack. J. Yan *et al.* used optical flow correlation to detect photo face spoofing attacks [9].

2.1.1.3. IMAGE QUALITY METHODS

Spatial liveness information is extracted from static images, with the expectation that this kind of method designs multiple features to capture the superimposed noise and illumination information of the spoof images. The spoof will have an image quality different from the real face, including sharpness, textures, or luminosity. These differences allow the method to distinguish the fake from the real image.
Another kind of spoof is the face morphing, which happens when an individual blends a face image with a target face image, resulting in a face with the characteristics of two different subjects. These spoof attacks strongly challenge face-verification systems, as they typically match two different identities.

One approach to this problem was proposed by S. Autherith and C. Pasquini, where they analyze the locations of important facial features identified in the image captured in the moment and the respective passport image. The goal is to capture changes in the facial geometry introduced by the morphing process [10].

Another study, from R. Raghavendra, K. B. Raja, and C. Busch, proposes a novel strategy based on facial micro-textures extracted from independent filters that are trained on real images [11].

The work developed by I. Medvedev *et al.* for ISR, in the area of face image verification, prevents face image spoofing with morphed faces by encoding the person facial features in a machine-readable code which is protected from face photo manipulation [12]. This work granted great results in the National Institute of Standards and Technology (NIST) Face Recognition Vendor Test (FRVT) challenge [13].

2.1.1.4. Deep learning-based methods

The fourth type of method is deep-learning-based methods, which use a CNN to learn discriminative features by considering face anti-spoofing as a binary classification problem. Not every kind of face spoofing attack can be detected from a single modality, so combining different modalities can solve several attack specific sub-problems. Although there has been significant progress in RGB images face anti-spoofing, only a few studies have tackled the multi-modal task.

L. Feng *et al.* introduced a method using multi-modal data with a CNN and a bottleneck feature fusion approach, obtaining near-perfect classification [14]. The results confirm the importance of a proper feature fusion strategy for multi-cues integration in face anti-spoofing problems.

H. Chen *et al.* developed an illumination-invariant method for anti-spoofing [15]. They proposed a two-stream convolutional neural network (TSCNN) which works on two complementary spaces: RGB space (original imaging space) and MSR (illumination-invariant space). The RGB space contains detailed facial textures yet is sensitive to illumination, opposing the MSR space, which is invariant to illumination and can effectively capture high-frequency information (discriminative for face spoofing detection) but contains less detailed facial information. Images from both spaces are supplied to the TSCNN to learn the discriminative features for anti-spoofing.

R. Shao *et al.* proposed a novel feature learning model [16] to determine discriminative deep dynamic textures for 3D mask face anti-spoofing and learn the spatial and channel-discriminability of these textures. The proposed strategy can adaptively weight the learned feature's discriminability from different spatial regions or channels, ensuring that more discriminative deep dynamic textures play more essential roles in face/mask classification.

Z. Yu *et al.* extended the central difference convolutional networks (CDCN) to a multi-modal version [17], aiming to obtain intrinsic spoofing patterns among the three modalities (RGB,

depth, and near-infrared). They also give a detailed study about single-modal based CDCN and introduce central difference into vanilla convolution to improve its representation and generalization capacity.

Sheng *et al.* also adopt a multi-stream CNN architecture called FaceBagNet [18]. They use patch-level images as inputs to improve their local representation ability and apply modality feature erasing operations to prevent overfitting and obtain more robust modal-fused features. H. Kuang *et al.* propose in 2019 a novel Multi-modal Multi-layer Fusion Convolutional Neural Network [19]. This CNN can use the different information provided by various modalities, which is based on a weight-adaptation aggregation approach. A multi-layer fusion model is used to compile the features from the different layers. Finally, a novel Average Binary Center loss function is proposed to enhance the contrast between the real faces and the spoofs.

Zhang *et al.* propose a three parallel stream based on a ResNet18 backbone, where the input of each stream corresponds to each modality face images, RGB, depth and IR [20]. The streams are concatenated and pass through to the last two residual blocks to the classification phase.

Similarly, A. Parkin *et al.* , in the ChaLearn Anti-Spoofing Challenge 2019 edition, also use multi-modalities as the CNN input but with fusion and excitation as a data fusion method [21]. This method used a modified network architecture from the baseline network [5]. The three modalities (RGB, Depth and IR) inputs are processed by separate channels and then pass through concatenation and fully-connected layers. The difference from the baseline method is in the use of aggregation blocks to group outputs from multiple layers of the network. A. Parkin *et al.* pre-train the network weights on different tasks, such as face and gender recognition, improving these networks separately on the training set of the CASIA-SURF dataset [5], splitting the training set into three groups according to different spoof attacks in the training subset. This division allows the CNN to increase its robustness to unknown spoof attacks. The channels' outputs are finally combined by averaging to generate results on the final validation and test sets.

Some methods specific to detecting morphed faces were also proposed, although they are not multi-modal.

C. C. Hsu et al. propose a new method for detecting fake images by using contrastive loss [22]. A reduced DenseNet is improved to a two-streamed siamese network structure allowing pairwise information as the input. The proposed shared fake feature network is trained using

pairwise learning to distinguish the features between the false and authentic images, and lastly, a classification layer evaluates the face images.

P. Zhou et al. also propose a two-stream network to detect modified face images [23]. The network is divided into a first stream based on GoogLeNet to detect changes in the face classification and a second stream to train a patch-based triplet network.

2.1.2. COMBINED MODALITIES FOR LIVENESS DETECTION

As previous mentioned, some modalities information is more reliable, having a larger impact on spoof detection methods. The RGB modality is not sufficient to provide a high level of security. So other image modalities, provided by different cameras, such as IR or Depth, capture different useful features (such as eye reflection, light distribution, face surface), making anti-spoofing models more reliable and secure. IR data measures the amount of heat radiated from a face, so Kim et *al*. use this information to differentiate the facial skin and mask materials, exploiting their reflectance [24]. The depth data can be used to classify a face as real or spoof since the depth map is usually different in a spoof face image from a real face image. One reason frequently stated for the superiority of the depth images over RGB images is that they are illumination invariant while the RGB facial appearance can be affected by illumination in various ways. The depth shape exists regardless of how it is illuminated. However this shape can change due to low lighting or when the detectors are saturated with high-intensity lighting [25].

Table 1 presents the results obtained in [5] for the the CASIA-Surf baseline method, according to all possible combinations of the three modalities, showing that the combination of the three modalities gets the best results.

Modality	APCER(%)	NPCER(%)	ACER(%)	TPR(%)		
				@FPR 10e-2	@FPR 10e-3	@FPR 10e-4
RGB	8.0	14.5	11.3	49.3	16.6	6.8
DEPTH	5.1	4.8	5.0	88.3	27.2	14.1
IR	15.0	1.2	8.1	65.3	26.5	10.9
RGB & Depth	4.3	5.6	5.0	86.1	49.5	10.6
RGB & IR	14.4	1.6	8.0	79.1	50.9	26.1
Depth & IR	1.5	8.4	4.9	89.7	71.4	24.3
RGB & Depth & IR	3.8	1.0	2.4	96.7	81.8	56.8

Table 1 - Effect on the number of modalities with the CASIA-SURF baseline method

The metrics used for the models validation will be explained with detailed information in chapter 5.

2.2. Facial Recognition

Recent technology allowing verification of real individual identity became available based in a field called biometrics, which is a technique for identifying people using a unique physiological characteristics, such as fingerprints, eyes, face or other behavioral characteristics.

The study of artificial neural networks (ANNs) has received significant research interest, both as predictors of non-linear models and pattern classification, requiring only representative training data instead of an accurate mathematical model of the process.

Several emerging applications demand the development of efficient and automated face recognition systems such as law enforcement, biometric authentication, and other commercial tasks.

Face recognition is a demanding research topic in computer vision and pattern recognition due to variations in different aspects such as poses, illumination, emotions and facial expressions. Furthermore, when the dimension of the face database increases, the recognition time becomes a significant limitation.

Face recognition analyzes the components of a person face images and it is generally designed to compute the similarity of facial images with high accuracy and low intrusiveness.

Although many researchers have worked on face recognition for multiple years, several challenges still need to be solved. Most face recognition methods have been created to perform correctly in controlled environments [26]. The methods working in uncontrolled environments are usually based on the texture and appearance descriptors formed by combining various local descriptors of the face image into a global descriptor. The local features based face recognition methods have gained approval instead of traditional methods [27], [28] because they are less sensitive to previously mentioned variations in the image.

2.2.1. Facial Recognition Methods

The accuracy of face recognition has been significantly improved using deep learning networks because of its ability to extract human faces deep features. We first analyze several methods that perform facial recognition using appearance-based methods and then methods using neural networks.

2.2.1.1. Appearance-based and Texture-based Methods

Appearance-based representation is based on recording various statistics of the pixels' values within the face image and these techniques are generally used in the subjects' identification through their facial images. Some of the popular techniques are linear discriminant analysis [29], fisher face [30], [31], and independent component analysis [32].
On the other hand, texture-based techniques usually vary in the type of texture representation used. One of these methods include the traditional Local Binary Pattern presented by T. Ojala, M. Pietikäinen, and D. Harwood [33].

Along these lines, Sun et al. propose to extract deep features from 25 cropped image patches with various scales and positions [34]. The dimension of the concatenated deep features is then reduced by Principle Component Analysis.

R. Shyam and Y. N. Singh present a state-of-the-art multi-modal biometric method for face recognition that combines the similarity scores of the unimodal modalities such as appearance-based and texture-based [35]. This method includes the fusion of technique in four possible combinations such as Eigenfaces and Local Binary Pattern (LBP), Fisherfaces and LBP, Organic and Augmented LBP (A-LBP), and Fisherfaces and A-LBP.

2.2.1.2. Deep learning-based Methods

Face recognition in the modern era heavily relies on recent advances in machine learning. In most of these approaches, the face representation is designed based on low-level facial image information. These techniques are often used to learn face descriptor discrimination by completing a classification task to evaluate image similarity. Deep convolutional networks allow for the rapid learning of discriminative face characteristics even from unconstrained images.

J. Liu *et al.* propose an approach for face verification and recognition combining a multi-patch deep CNN and metric learning , allowing the extraction of low dimensional but highly discriminative features [36]. The authors use a network structure with nine convolution layers and a Softmax layer at the end for supervised multiclass learning. This method can handle cases with variant poses, occlusions, and expressions, and while the number of identities and faces per identity in training data increases, the performance also improves.

R. Ranjan *et al.* present a multi-objective algorithm for simultaneous face detection and alignment, pose estimation, gender recognition, smile detection, age estimation, and face recognition using a single deep CNN [37]. This method divides the tasks into subject-independent tasks (face detection, key points localization, pose estimation, and smile prediction) and subject-dependent tasks (age estimation, gender prediction, and face recognition).
The authors share the parameters of the lower layers of the CNN for the different tasks to produce a generic face representation followed by the task-specific layers, regularizing the shared parameters of CNN and collaborating among the tasks.

C. Han *et al.* propose a CNN structure with a contrastive convolution [38], which explicitly focuses on the obvious characteristics between the two faces.
The authors present a kernel generator module to generate personalized kernels of the faces. The CNN can then perform convolutions with those contrastive kernels and extract contrastive features of two faces for the similarity calculation.

J. Yang *et al.* present a Neural Aggregation Network (NAN) for face recognition [39]. The proposed network uses face videos to create feature representations and it is composed of two modules. The feature embedding module is a deep CNN to map face images to a feature vector. The aggregation module consists of two attention blocks that aggregate the feature vectors to form a single feature, causing the aggregation to be invariant to the image order. This network automatically learns to choose high-quality face images while opposing low-quality ones such as occluded, blurred, and improperly exposed faces.

2.2.2. Facial Recognition Pipelines

Typical network architectures for facial recognition with liveness detection use one of two types of pipelines: series/cascaded pipeline or parallel pipeline.

Both pipelines need to run the same image two times, one for liveness detection and one for face recognition, but the series pipeline has the advantage of lower computation costs, since it runs the liveness detection task before the recognition task, only at a cost of taking twice the time of the parallel pipeline.

The usage of either of the pipelines depends on the purpose of the system and the resources available.

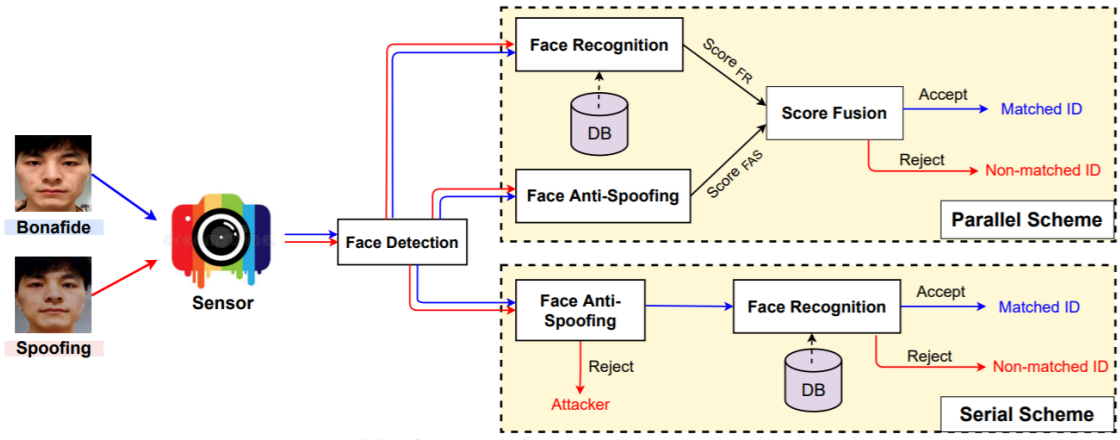

Figure 2 – Series and parallel anti-spoofing and facial recognition pipelines [40]

2.3. Fusion methods

There are several data fusion methods to use when integrating multi-modal data. The multi-modal fusion purpose is to integrate the multiple data representations into one representation that can be used to classify them as real or spoof attacks. The advantages of using multi-modal fusion methods are robustness in the resulting prediction, collection of new information that is not visible in single modalities, and a multi-modal system can still run when one of the modalities is missing.

Data fusion can be categorized as:

- **Early fusion strategy** - Local features are obtained from the same patches and then fused before encoding, making the training easier than late and halfway fusions. The fusion happens in the starting layers
- **Halfway fusion strategy** - Takes advantage of both methods mentioned before by merging the different modalities in the middle stage. The fusion happens near the end of the network but before the fully connected layers

- **Late fusion strategy** - The image representations are calculated for each feature and fused after. The fusion happens in the last layers, typically on the fully connected layer

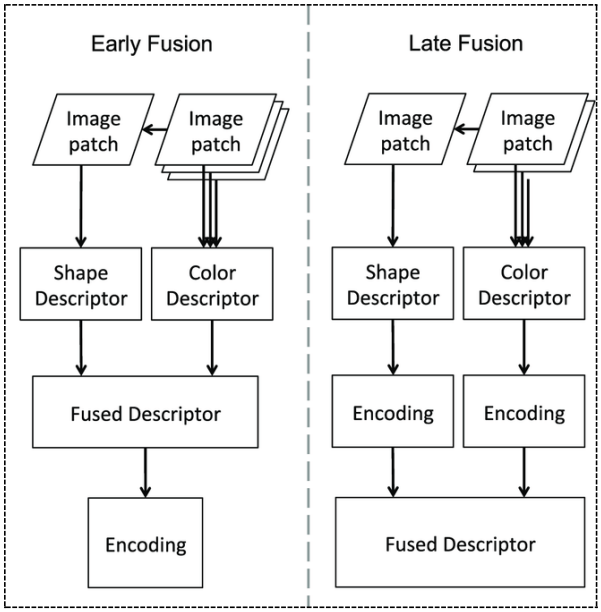

Figure 3 - Early and Late Data Fusion

2.4. Loss Functions

Loss functions play an important role in deep feature learning. The Softmax loss was used to learn face features in pioneering works [41], [42], however this type of loss only learns separable features that are not discriminative enough.

To address this, some methods combine Cross-Entropy loss with contrastive loss [34] or center loss [43] to enhance the features discrimination power. However, center loss only explicitly encourages intra-class compactness. Contrastive loss [34] cannot constrain on each individual sample, and thus requires carefully designed pair mining procedure, which is both time-consuming and hurts the systems performance. Compared to original Softmax loss, the features learned by the modified Softmax losses are angularly distributed, but not necessarily more discriminative, and to this end the Large Margin Softmax Loss (LMCL) [44] and the Angular Softmax (A-Softmax) [45] have been proposed.

- **Softmax loss**

The Softmax function takes a vector of real numbers as input and normalizes it into a probability distribution according to John S. Bridle [46]. Before applying Softmax, some vector

components could be negative or greater than one and might not sum to 1. However, after applying Softmax, each component will be in the interval [0,1], and the components will add up to 1 to be interpreted as probabilities.

The decision boundary is defined by the Softmax loss (also known as Cross Entropy Loss) as $\|W_1\|\cos(\theta_1) = \|W_2\|\cos(\theta_2)$ and the Softmax loss is defined as:

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

$$\begin{aligned} L_i &= -\log\left(\frac{e^{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_j e^{\boldsymbol{W}_j^T \boldsymbol{x}_i + b_j}}\right) \\ &= -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i}) + b_{y_i}}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_{j,i}) + b_j}}\right) \end{aligned}$$

Its boundary depends on the magnitudes of weight vectors and cosine of angles, which results in an overlapping decision area in the cosine space, as seen in Figure 4.

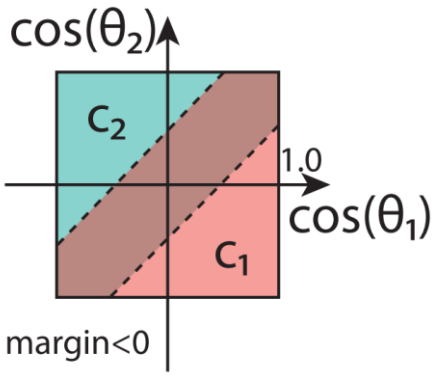

Figure 4 - Softmax Loss decision margins (gray area) for a binary-class scenario [44]

- **Normalized Softmax loss (NSL)**

The NSL normalizes the weight vectors W1 and W2 such that they have constant magnitude 1, which results in a decision boundary given by $\cos(\theta_1) = \cos(\theta_2)$.

This loss function can correctly classify testing samples in the cosine space by removing radial variations, with margin = 0 (view Figure 5). However, it is not robust to noise because there is no decision margin.

The NSL is defined as:

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

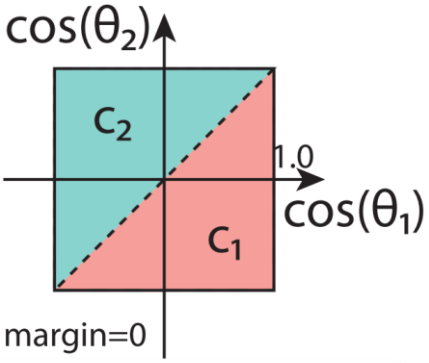

Figure 5 - Normalized Softmax Loss decision margins for a binary-class scenario [44]

- **Large Margin Cosine Loss**

The LMCL defines a decision margin in cosine space by:

$$C_1 : \cos(\theta_1) \geq \cos(\theta_2) + m,$$
$$C_2 : \cos(\theta_2) \geq \cos(\theta_1) + m.$$

Therefore, $cos(\theta_1)$ is maximized while $cos(\theta_2)$ is minimized for C1 (similarly for C2) to perform the large-margin classification.

The modified loss can be formulated as:

$$L_{lmc} = \frac{1}{N}\sum_i -\log\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{s\cos(\theta_{j,i})}}$$
$$W = \frac{W^*}{\|W^*\|},$$
$$x = \frac{x^*}{\|x^*\|},$$
$$cos(\theta_j, i) = {W_j}^T x_i,$$

In Figure 6 we can see a clear margin between the two classes suggesting that the LMCL is more robust than the NSL because a small perturbation around the decision boundary is less likely to classify incorrectly.

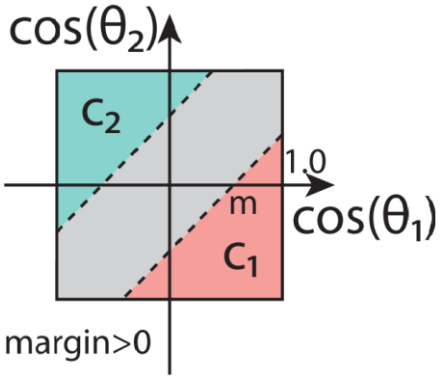

Figure 6 - Large Margin Cosine Loss decision margins (gray area) for a binary-class scenario [44]

- **Angular Softmax Loss**

The A-Softmax loss function approach is similar to the LMCL. The objective of this loss is also to control the decision boundary by introducing an integer m. This value m has such lower bounds that the A-Softmax loss can approximate the learning task to have a minimal inter-class distance larger than the maximum intra-class distance.
Since we used this loss function, we will explain this method in detail in section 3.6.

2.5. Datasets

The majority of the existing face anti-spoofing datasets consist of one or two modalities, being the RGB images the most frequent. These datasets have two general constraints: the limited number of subjects and the use of a single modality. Spoof attack techniques are continuously upgraded, especially with the development of 3D and silicone masks. These new types of presentation attacks are more realistic than traditional 2D attacks, revealing the disadvantages of only using a single modality.
Similarly, anti-spoofing techniques are also continuously upgraded. New sensors are introduced, such as depth cameras or infrared cameras, to help tackling the drawbacks revealed by the new spoof attacks.

Y. Zhang *et al* create an anti-spoofing dataset [3] with 625,537 pictures of 10,177 different subjects, which is significantly larger than any other anti-spoofing dataset. The spoof images are captured in two environments with four different illumination conditions, and there are ten spoof attack types. Even though this is an excellent dataset for single modality anti-spoofing, it lacks the abilities of the Depth and IR modalities.

Y. Liu *et al* develop the SiW dataset which provides live and spoof videos from 165 different subjects [4]. There are 8 live and 20 spoof videos for each subject, with 6 different attack types, totaling 4620 videos. This dataset is also a single modality dataset.

A new dataset was created by S. Zhang *et al.*, the CASIA-SURF Dataset [5], to overcome these difficulties. This dataset includes the standard RGB and IR modalities but also includes a third one, Depth. The CASIA-SURF consists of 1000 subjects with 21000 videos with the three modalities. The authors also provide extensive evaluation metrics, evaluation protocols and training/validation/testing subsets, developing a new benchmark for face anti-spoofing,

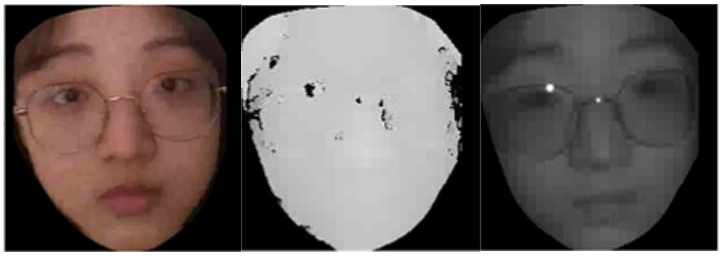

Figure 7 - Example images from the CASIA-SURF Dataset [20]

Another dataset CASIA-SURF CeFA [47] was created by A. Liu *et al.*, that covers 3 ethnicities, 3 modalities (RGB, IR and Depth), 1607 subjects, and 2D and 3D attack types. This dataset provides all the information needed to train, test and validate any liveness detection method.

2.6. CASIA-SURF CHALLENGE

The CASIA-SURF challenge [48] is a Multi-modal Face Anti-spoofing Attack Detection Challenge with two editions, Edition 2019 and 2020. This challenge consists of detecting the liveness of pictures from 3 modalities (RGB, IR and Depth), using the CASIA-SURF Dataset [20]. This challenge involved more than 300 teams with 13 qualifying for the final round.

For the 2019 competition, a strong baseline for approaching Face Anti-spoofing Attack Detection was proposed in by S. Zhang *et al.* in [5]. The proposed solution considered the face anti-spoofing problem as a binary classification problem (spoof vs. real) and operated experiments based on the ResNet-18 classification network [49]. To use the different features of all the modalities, it was proposed the Squeeze and Excitation fusion method [50]. This method uses the “Squeeze-and Excitation” block to improve the different modalities’ ability to be represented. This improvement is due to performing a feature re-weighting and selecting the more informative channel features while suppressing the less useful ones for each modality.

Article	Method	Model	Pre-trained data	Modality	Pre-process	Additional FAD dataset	Fusion and Loss function
Baseline [5]	Features fusion	ResNet-18	No	RBG Depth IR	Resize Image augmentation	No	SoftmaxWithLoss

H. Kuang mmf CNN [19]	Features fusion	ResNet-34	No	RBG Depth IR	Transfer color space	No	Features fusion SoftmaxWithLoss
A. Parkin Agg CNN [21]	Fine-tuning Ensembling	ResNet-34 ResNet-50	Casia-WebFace AFAD-Lite MSCeleb1M Asian dataset	RBG Depth IR	Resize	No	Squeeze and Excitation Fusion Score fusion SoftmaxWithLoss

Table 2 - Methods used in the Multi-modal Face Anti-spoofing Attack Detection Challenge at CVPR2019

Article	**FP**	**FN**	**APCER (%)**	**NPCER (%)**	**ACER (%)**	**TPR(%)**		
						@FPR 10e-2	**@FPR 10e-3**	**@FPR 10e-4**
[5]	1542	177	3.8308	1.0138	2.4223	96.7464	81.8321	56.8381
[19]	825	30	2.0495	0.1718	1.1107	99.5131	97.2505	89.5579
[21]	3	27	0.0074	0.1546	0.0810	99.9885	99.9541	99.8739

Table 3 - Multi-modal Face Anti-spoofing Attack Detection Challenge at CVPR2019 final results

Having in consideration all the aspects and relevance of the CASIA-Surf Dataset and the results achieved by A. Parkin in the 2019 ChaLearn Anti-Spoofing Challenge, all the work in this dissertation is highly inspired by their efforts and accomplishments.

3

Methodology

In this chapter, we dedicate special attention to the used state-of-art methods and the methodology extensions adopted for the purpose of this dissertation. We start by presenting some basic CNN concepts [51]. Also, we present a baseline network architecture used for the ChaLearn Anti-Spoofing Challenge 2019, followed by the challenges' winner paper network architecture, which will be our baseline. Then we explain the modules used in said baseline architecture, the Squeeze and Excitation and Channel Attention modules. In contrast to the loss function used in the baseline, we replace the Original Softmax Loss with the Angular Softmax Loss. Apart from the incorporation of this new loss we also dedicate some attention to several extensions to the baseline architecture.

3.1. Basic Blocks

Generally, the architecture of a CNN is composed of a sequence of 5 essential layers:

- **Convolution layer -** An input image is convolved with a learnable kernel. Each neuron processes some data by taking the dot product between the kernel values and image pixels, resulting in a feature map.

- **Activation Layer** - Introduces non-linearity in the system by mapping the generated features into non-linear values. Some popular non-linearity functions are the Rectified Linear Unit (ReLU), tanh, PreLU, and sigmoid.

- **Pooling layer** - Extracts relevant features of an image by down-sampling the output of the activation layer. The objective of a pooling layer is to identify the presence of a feature rather than its exact location while reducing the spatial dimensionality of the input image, decreasing the computational complexity. The pooling layer can be of type Max-pooling, Global-pooling or average pooling.

- **Drop-out Layer** - Assigns a zero value to a random set of activation values from the previous layer, generalizing the model over the training data and reducing the overfitting in the model.

- **Fully connected (FC) Layer** - Every neuron in the previously hidden layer is connected to every neuron of this layer. The fully connected layers learn a function between the high-level features given as an output from the convolutional layers.

3.2. CASIA-SURF BASELINE ANTI-SPOOFING ARCHITECTURE NETWORK

Z Zhang *et al.* [20] introduced the CASIA-SURF dataset and a baseline method for the multi-modal face anti-spoofing task. They view the face anti-spoofing problem as a binary classification task (fake versus real) and base their network architecture on the ResNet-18 and ResNet-34 [49]. The ResNet classification network consists of five convolutional blocks (res1, res2, res3, res4, and res5), a global average pooling layer (GAP), and a Softmax layer for label classification.

They adapt their network structure to fuse the multi-modality (RGB, Depth, and IR) present in the CASIA-SURF Dataset using a multi-stream architecture with three subnetworks. Each subnetwork learns different modality features, and later shared layers are fused to learn joint representations and perform combined decisions.

The naive halfway fusion is a method that combines the subnetworks of different modalities in a later stage by feature map concatenation, allowing the network to perform classification.

In order to improve the naive halfway fusion, they also implement a Squeeze and Excitation (SE) fusion to make full use of the characteristics between different modalities. The SE module joins a branch to obtain the channel-wise weights for each modality, re-weights the input features, and combines these re-weighted features. This fusion allows the selection of the more informative channel features while suppressing less valuable features from each modality.

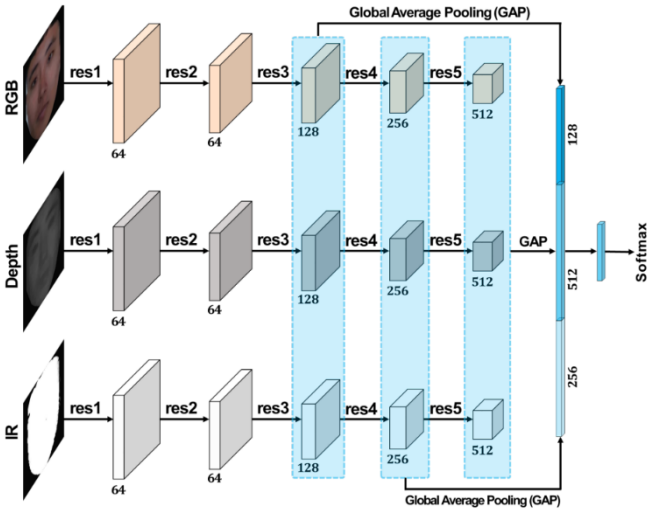

Figure 8 - Zhang et al. proposed network architecture [20]

3.3. Aggregation Multi-Modal Anti-Spoofing architecture Network

A. Parkin *et al.* [21] network architecture is also based on the ResNet-34 backbone with SE modules. Their network follows the method described in the baseline network but differs in some aspects. The authors improve the model with aggregation blocks at each feature level. Each aggregation block takes features from the corresponding residual blocks and the previous aggregation block, making the model fitted to find inter-modal correlations at a fine and coarse level.

They also increase robustness to new attacks by splitting the training data into three folds, using two different attack types for training and the third attack type for validation. This method prevents the network from overfitting and allows it to detect attack types not in the dataset at run time.

Fine-tuning network parameters pre-trained on various source tasks leads to different results on the task at hand. The authors use multiple backbone ResNet architectures and losses for initial tasks to increase variability and multiple datasets designed for face recognition and gender classification to create good initialization for the face anti-spoofing networks.

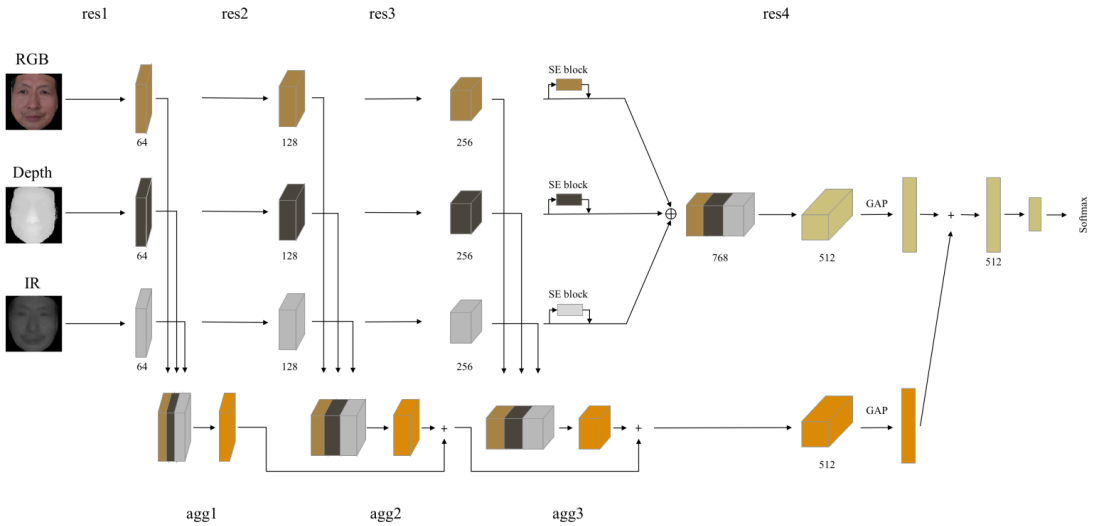

Figure 9 - A. Parkin et al. proposed network architecture [21]

3.4. Squeeze and Excitation Module

J. Hu *et al.* introduce the Squeeze and Excitation (SE) blocks [50] to enhance the features produced by a typical convolutional network.

The SE block models the interdependencies between the channels of its convolutional features. This block allows the network to perform feature recalibration, learning to use global information to emphasize useful features and suppress the useless ones through weights like an attention mechanism.

The input features pass through a squeeze operation to aggregate global information per channel for the whole image. The outputs are then passed through an excitation operation to get the final weights for each channel. Finally, the weight vectors are reshaped to match the number of the feature maps.

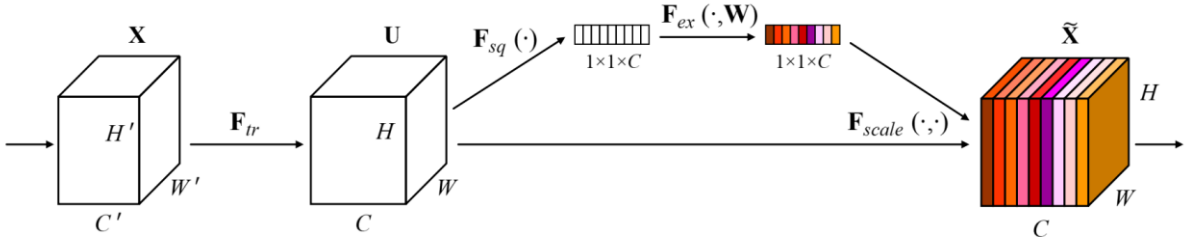

Figure 10 - Squeeze-and-Excitation block [50]

Each SE block uses a global average pooling operation in the squeeze phase and two small FC layers and one ReLU and sigmoid activation layer in the excitation phase.

In earlier layers, it stimulates informative features without considering classification, strengthening the shared low-level representations. In later layers, the SE blocks become increasingly specialized and respond to different inputs in a highly class-specific way.

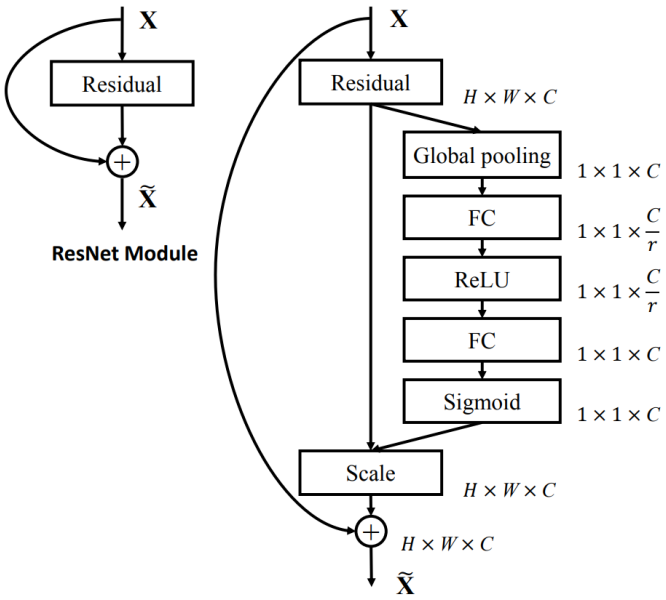

Figure 11 - Scheme of the original Residual module (left) and the SE ResNet module (right) [50]

3.5. Convolutional Block Attention Module

The Convolutional Block Attention Module (CBAM) [52] is a simple and effective attention module designed for feed-forward CNNs. This module sequentially creates attention maps for the channel and spatial dimensions and then multiplies them by the input feature map for adaptive feature improvement.

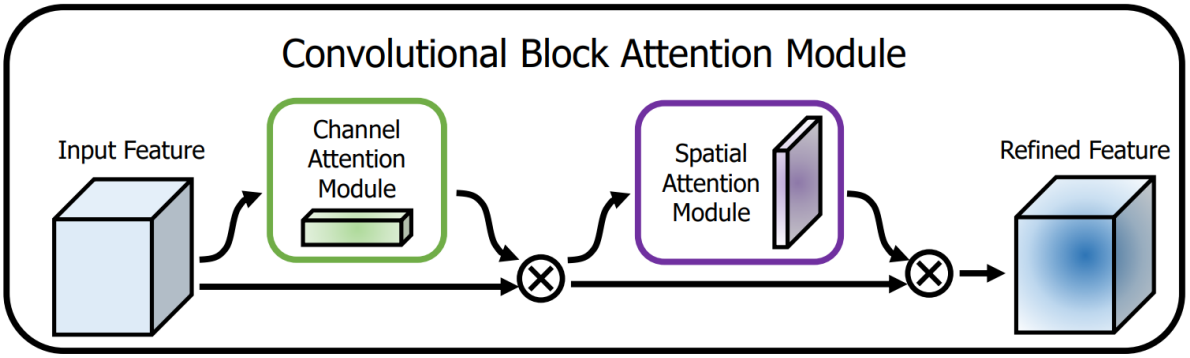

Figure 12 - Convolutional Block Attention Module [52]

S. Woo *et al.* create a channel attention map using the features inter-channel relationship. Each feature map channel is considered a feature detector, so channel attention focuses on what is meaningful given an input image. The spatial dimension of the input feature map is squeezed by average-pooling to compute the channel attention efficiently.

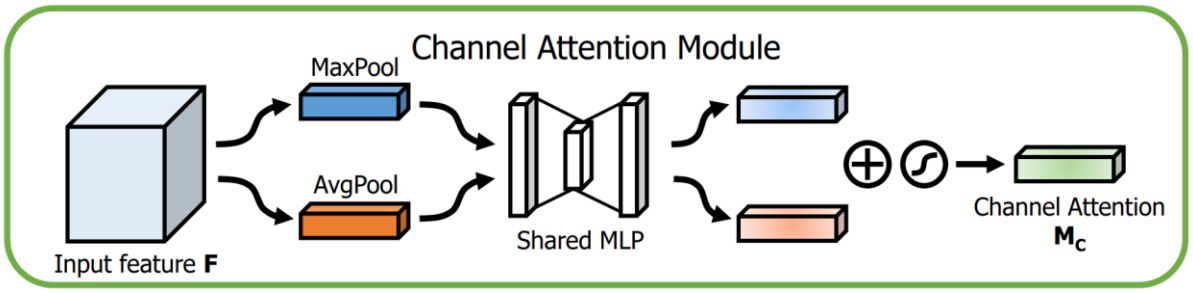

Figure 13 - Channel Attention Module [52]

The spatial attention focuses on where is an informative part, contrary to channel attention that focuses on what is informative.

Average-pooling and max-pooling are applied along with the channel axis feature descriptor, which is then concatenated to generate a useful feature descriptor.

A convolution layer is applied on the concatenated feature descriptor to generate the spatial attention map, which encodes where to highlight or suppress.

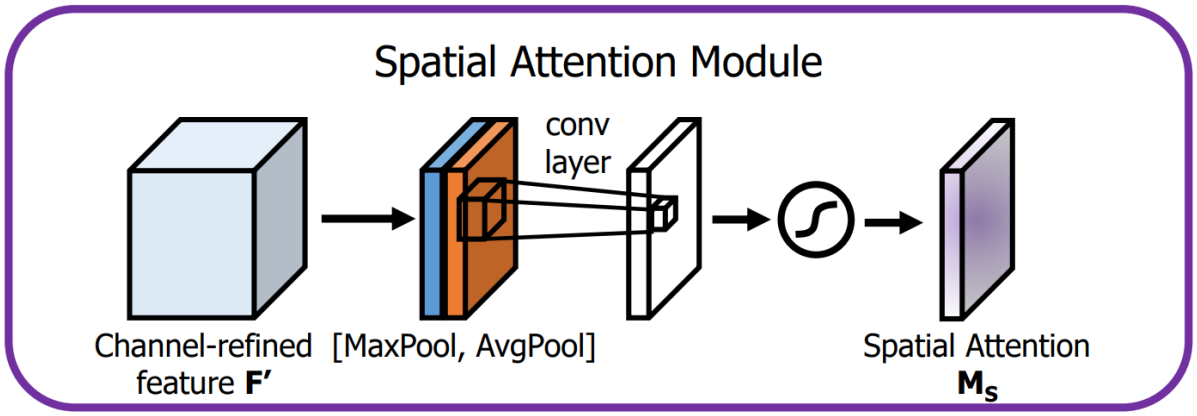

Figure 14 - Spatial Attention Module [52]

3.6. A-Softmax Loss

After analyzing the Softmax loss, W. Liu *et al.* [45] learn that decision boundaries can significantly affect the feature distribution, so they manipulate the decision boundaries to produce an angular margin. Furthermore, by optimizing A-Softmax loss, the decision regions become more separated, simultaneously enlarging the inter-class margin and compressing the intra-class angular distribution, resulting in a clear geometric interpretation.

A-Softmax loss allows CNN's to learn face features with geometrically interpretable angular margin, improving the Softmax loss by introducing an extra margin, such that its decision boundary is given by:

$C1: \cos(m \cdot \theta_1) \geq \cos(\theta_2)$,

$C2: \cos(m \cdot \theta_2) \geq \cos(\theta_1)$, where C1 and C2 are two different classes.

Consequently, for C1, it requires $\theta_1 \leq \frac{\theta_2}{m}$, and similarly for C2, $\theta_2 \leq \frac{\theta_1}{m}$. However, the margin is smaller for similar classes C1 and C2, therefore having a smaller angle between W1 and W2, meaning the margin decreases alongside θ and disappears when $\theta = 0$, as seen in Figure 15.

The proposed A-Softmax loss is formulated as:

$$L_{ang} = \frac{1}{N} \sum_i -\log\left(\frac{e^{\|x_i\|\psi\left(\theta_{y_i,i}\right)}}{e^{\|x_i\|\psi\left(\theta_{y_i,i}\right)} + \sum_{j \neq y_i} e^{\|x_i\| cos(\theta_{j,i})}}\right)$$

$$\psi\left(\theta_{y_i,i}\right) = (-1)^k \cos\left(m\theta_{y_i,i}\right) - 2k$$

$$\theta_{y_i,i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right], \qquad k \in [0, m-1], \qquad m \geq 1$$

where $e^{\|x_i\|\psi\left(\theta_{y_i,i}\right)}$ is the probability of belonging to the class and $\sum_{j \neq y_i} e^{\|x_i\| cos(\theta_{j,i})}$ is the sum of probabilities of not belonging to the class.

A-Softmax loss requires W = 1, b = 0, causing the prediction to only depend on the angles θ between W and the sample x. The variable *m* is an integer that controls the size of the angular margin.

To facilitate gradient computation and backpropagation, the authors replace $\cos(\theta_{j,i})$ and $\cos(m \cdot \theta_{yi,i})$ with expressions only containing W and x, using the definition of cosine and multi-angle formula. Without θ, we can compute derivatives for x and W, similar to Softmax loss.

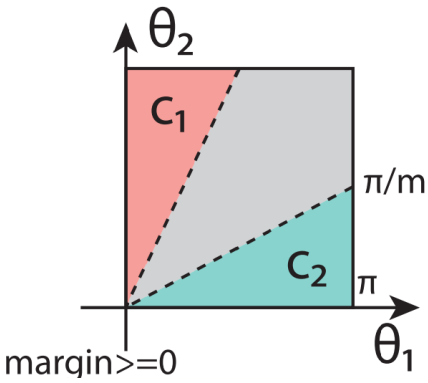

Figure 15 - SphereFace Loss decision margins (gray area) for a binary-class scenario [44]

Based on the annealing optimization strategy for A-Softmax loss on the annex G in [45] we get

$$f_{y_i} = \frac{\lambda \cdot \|x_i\| \cdot \cos(\theta_{y_i}) + \|x_i\| \cdot \psi(\theta_{y_i})}{1+\lambda} = \frac{\|x_i\| \cdot \cos(\theta_{y_i}) + \lambda \cdot \|x_i\| \cdot \cos(\theta_{y_i}) + \|x_i\| \cdot \psi(\theta_{y_i}) - \|x_i\| \cdot \cos(\theta_{y_i})}{1+\lambda}$$

$$= \frac{(1+\lambda) \cdot \|x_i\| \cdot \cos(\theta_{y_i}) + \|x_i\| \cdot \psi(\theta_{y_i}) - \|x_i\| \cdot \cos(\theta_{y_i})}{1+\lambda}$$

$$= \|x_i\| \cdot \cos(\theta_{y_i}) + \frac{\|x_i\| \cdot (\psi(\theta_{y_i}) - \cos(\theta_{y_i}))}{1+\lambda}$$

So the A-Softmax Loss becomes $L_{ang} = \frac{1}{N}\sum_i -\log(\frac{f_{y_i}}{\sum_j e^{f_j}})$ where $f_{y_i} = \|x_i\| \cdot \cos(\theta_{y_i}) + \frac{\|x_i\| \cdot (\psi(\theta_{y_i}) - \cos(\theta_{y_i}))}{1+\lambda}$.

3.7. Conditional A-Softmax Loss

We propose a Conditional A-Softmax Loss for the facial recognition task. To implement this loss function we change the original A-Softmax Loss to include a probability decision component. The decision task is to choose which samples are more or less viable for the recognition task during the training phase, according to the liveness scores for each sample.

The SphereFace loss starts by getting the embeddings from a fully connected layer following the formulas from [45]. With these embeddings we perform the probability decision and apply the annealing optimization strategy that the authors provide in their open-source code[3][4], previously mentioned in 3.6, and finally apply the logarithmic Softmax to the output, resulting in the loss for the backpropagation process.

[3] https://github.com/wy1iu/sphereface#note
[4] https://github.com/wy1iu/LargeMargin_Softmax_Loss#notes-for-training

This decision works by taking a subject's liveness probability and filling a one-hot vector with the mean of the remaining probability. The decision ensures the facial recognition trains with faces where the liveness probability is higher. If the liveness probability is closer to 1, the correct class gets a higher probability while the rest get a probability closer to 0 (Figure 18). If the liveness probability is closer to 0, all the classes will have the same probability (Figure 19), meaning the network will train with any class when the liveness detection results in a spoof, so it does not disturb the facial recognition for each specific class with a spoof face image, and with the correct class for the facial recognition when the liveness detection results in a real detection.

```
22      def forward(self, input_x, target, probability=1):
23  >       """…
31          |       You, 2 months ago • New Code formating
32          cos_theta, phi_theta = input_x
33          target = target.view(-1, 1)
34
35          self.iter += 1
36
37          one_hot = torch.ones_like(cos_theta) * (1-probability)/(cos_theta.shape[1]-1)  # fill one-hot with probability that remains
38          one_hot.scatter_(1, target, probability)  # scatter the correct liveness probability on the corresponding class
```

Figure 16 - Probability for loss function code segment

```
217 # Liveness correct probability for Recognition loss
218 if self.train_flag and self.loss_fnc == 'sf':
219     liveness_probability = F.softmax(model_output[0][0].detach())  # Liveness Probability . liveness sphereface outputs 2 values
220     liveness_correct_probability = liveness_probability.gather(1, labels[0].view(-1,1))  # Get the values according to label. If value is wrong, probability tends to 0, if correct tends to 1
221
```

Figure 17 - Probability for loss main function code segment

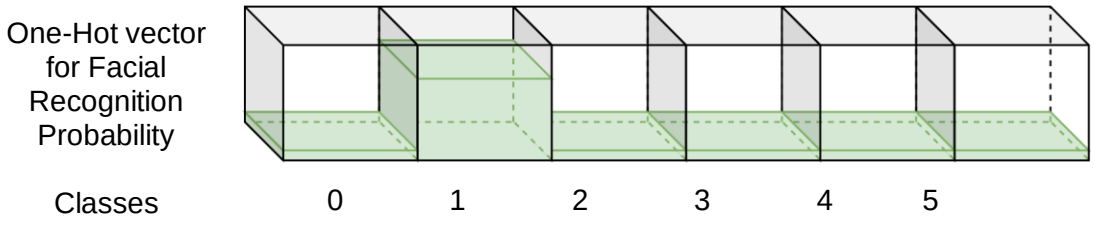

Figure 18 - High liveness probability for real face

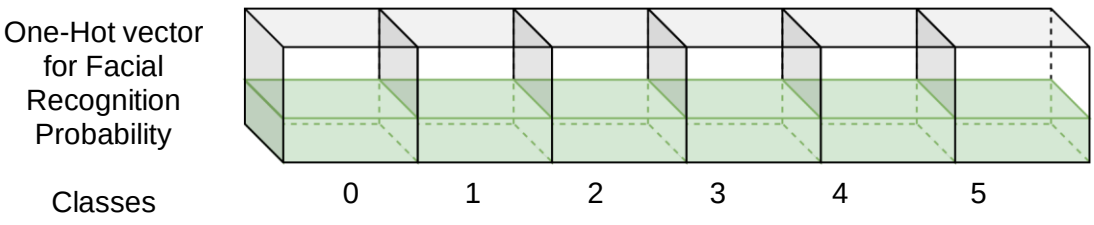

Figure 19 - High liveness probability for spoof

When the network starts the training process it might train the facial recognition with some wrong samples, but as the training process advances, the liveness detection improves and the wrong samples fed to the network for facial recognition are reduced.

When the samples are identified as a real subject in the liveness detection task, the facial recognition process trains the network with the model output with more confidence, while training with less confidence when the liveness detection task results in a spoof.

The new loss function is calculated as follows:

$$L_{cond\_ang} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{q_i \cdot \|x_i\| \cdot \psi\left(\theta_{y_i,i}\right)}}{e^{q_i \cdot \|x_i\| \cdot \psi\left(\theta_{y_i,i}\right)} + \sum_{j \neq y_i} e^{q_j \cdot \|x_i\| \cdot cos(\theta_{j,i})}}\right)$$

$$\psi\left(\theta_{y_i,i}\right) = (-1)^k \cdot \cos\left(m\theta_{y_i,i}\right) - 2k$$

$$\theta_{y_i,i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right], \qquad k \in [0, m-1], \qquad m \geq 1$$

$$q_j = \frac{1 - q_i}{total\ classes - 1}$$

Where q_i is the probability of the right class and q_j is the probability for the other classes. Using the same method adopted by the A-Softmax Loss, we can replace f_{y_i} in $L = \frac{1}{N}\sum_i -\log(\frac{f_{y_i}}{\sum_j e^{f_j}})$ by $f_{y_i} = \|x_i\| \cdot \cos(\theta_{y_i}) + \frac{q_i \cdot \|x_i\| \cdot \left(\psi(\theta_{y_i}) - \cos(\theta_{y_i})\right)}{1+\lambda}$.

3.8. Multi branch Loss

We also propose a multi-branch loss, to improve the results faster by refining the model on each branch separately (Figure 20). Each branch loss is calculated before the modalities fusion and added to the overall loss calculated after the fusion and aggregation, and each facial recognition loss is conditioned by the corresponding liveness detection value. The multi-branch loss is calculated as follows:

$$Loss_{final} = Loss_{Liveness} + Loss_{Recognition} + Loss_{multi}$$

$$Loss_{multi} = Loss_{RGB} + Loss_{IR} + Loss_{Depth}$$

$$Loss_{modality} = Loss_{Liveness_{modality}} + Loss_{Recognition_{modality}}, \qquad modality = RGB, IR\ and\ Depth$$

Where the recognition loss and the liveness loss can be a normal Cross Entropy Loss, A-Softmax Loss or Conditional A-Softmax Loss.

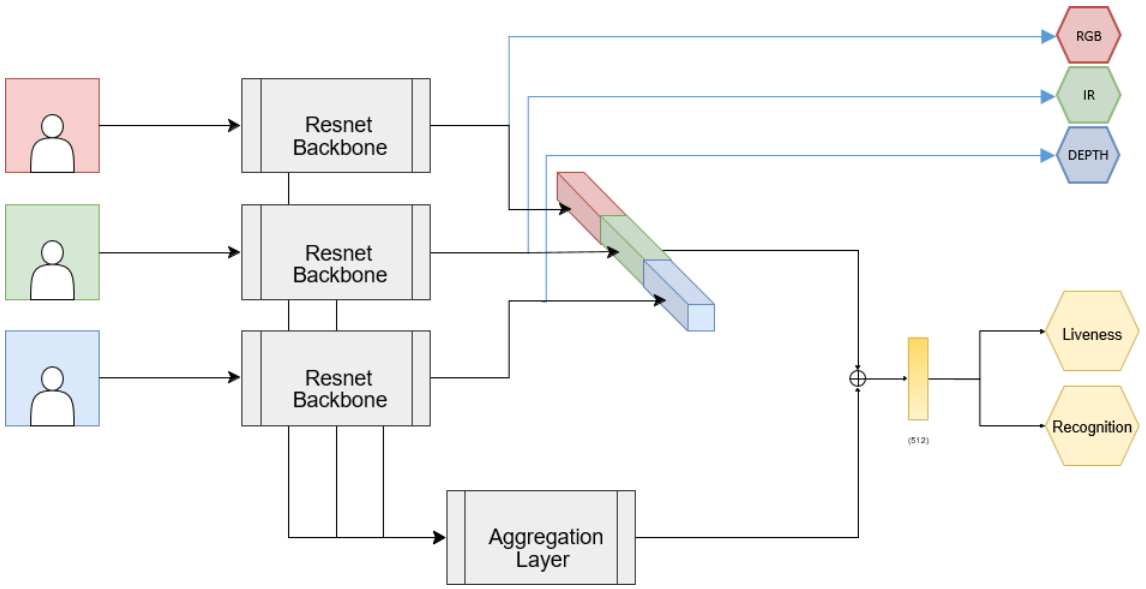

Figure 20 - Multi-branch Loss network architecture

4

Implementation Details

In this chapter, we will present the methodology implementation details for the CASIA-SURF Dataset and the dataset we created. We will also explain the data acquisition and real-time implementation details. Finally, we will present the results obtained after training with the different datasets.

4.1. Hardware

4.1.1. Jetson Nano

In order to create a more versatile and mobile facial biometric system we migrate to a Jetson Nano Board and create a script to install all the packages needed to run the app in this new board.

The NVIDIA Jetson Nano Developer Kit [5] is a small and powerful computer that allows the user to run multiple neural networks in parallel for image classification, object detection, segmentation, speech processing, or other applications.

This board is supported with the NVIDIA JetPack SDK [6] and, with a memory of 4 GB, is ideal for training and deploying artificial intelligence software.

[5] https://developer.nvidia.com/embedded/jetson-nano-developer-kit
[6] https://developer.nvidia.com/embedded/jetpack

Figure 21 - Jetson Nano Developer Kit

4.1.2. REALSENSE CAMERA

After training the model with the CASIA-SURF Dataset [5] and the SphereFace Loss [45] assuring the network was ready for different data we decided to use a camera with multiple modalities, i.e., RGB, near Infrared and Depth, to acquire our own images . The camera used is the Realsense D435[7], which has an RGB module and a depth stereoscopic module. It has a max RGB frame resolution of 1920 × 1080 and Depth output resolution up to 1280 × 720 and an ideal image range of 0.3m - 3m.

Intel manufactures this camera, and its use is ideal for fast-moving applications, enabling the development of next-generation sensing solutions that can understand and interact with their surroundings intended for developing purposes. The use of this camera is more accessible due to several programming languages wrappers available, such as Python, and various features such as On-Chip Calibration in Seconds.

Figure 22 - Intel Realsense D435

[7] https://www.intelrealsense.com/depth-camera-d435/

4.2. Software

All code was implemented in Python Language. Pytorch was used for the CNN architecture implementations; Qt for the graphical user interfaces; and pyrealsense2 and the realsense SDK to communicate with the camera.

4.2.1. Liveness Detection

Our system has two major tasks: anti-spoofing and facial recognition, which take part simultaneously, using the same weights and activations (for the detailed image see page 24). In the task of anti-spoofing, a feature vector (or embedding) is extracted from an input face image of size “224x224” using a CNN. For this task, the classification can either be live or spoof.

This task is adapted from the winning open-source code [21] used for the ChaLearn Anti-Spoofing Challenge 2019 [48] and is trained with the CASIA-SURF Dataset [5] to verify this state of the art face anti-spoofing method.

The network has three streams based on a ResNet network as a backbone, one for each modality, and an aggregation layer that collects information from the first convolutional block until after the modalities fusion (View Figure 23).

Each ResNet backbone can have multiple block numbers that consist of residual layers and the Squeeze and Excitation [50] and channel attention [52] blocks.

For the channel attention we apply separately a max pooling and an average pooling layer to the input tensor, followed by a fully connected and a ReLU activation layer for each. Then we sum the output of these layers and apply again a fully connected and a sigmoid activation layer. Finally we multiply the original input tensor with the output of the sigmoid layer.

In the Squeeze and Excitation block we apply a convolutional layer and ReLU activation layer twice and sum the output with the original tensor input.

The aggregation layer consists of a concatenation process, followed by a max pool and a normal convolution, i.e., for a stream with size 64, we concatenate the 3 modalities, resulting in a tensor of size 3 * 64 = 192. After the stream fusion, we perform a max pool operation followed by a normal convolution with tensor input size 64 and output size 128, which will be added element wise to the next aggregation block tensor.

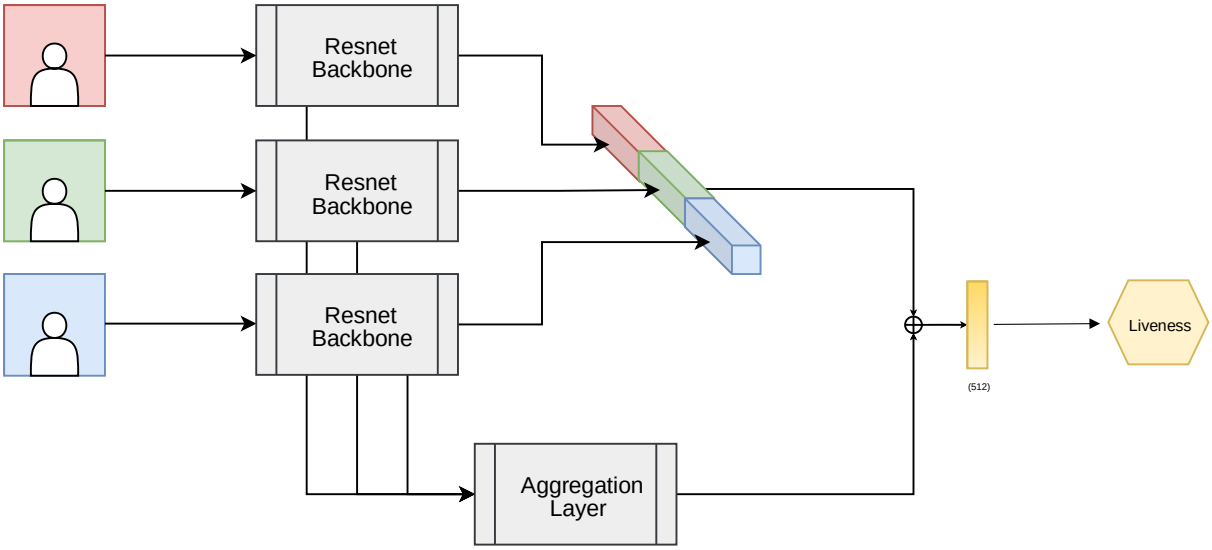

Figure 23 - Simplified proposed pipeline scheme

4.2.2. Facial Recognition

We extended the liveness detection task also to perform facial recognition. In a first stage, we assume the face recognition task is a closed-set task, but we intend to upgrade the system to perform as an open-set task in the future. The face could belong to someone already registered on the system or a new person not registered before. The proposed pipeline is described in Figure 24.

In order to also use the CASIA-SURF Dataset [5] for the facial recognition task, we had to adapt the dataset and lists to be able to use facial recognition simultaneously with the anti-spoofing system, using the same method but changing the size of the features extracted. Since the facial classification is done with the same descriptors as the liveness classification, we extended the anti-spoofing task by also using a Softmax loss for facial recognition to get the class with the best output. After we perform the loss operation for both tasks we add them as $final\ loss = liveness\ loss + recognition\ loss$ and perform the gradient descent from the final loss.

All code is open-source at GitHub[8].

[8] https://github.com/AndreGraca98/ASFR

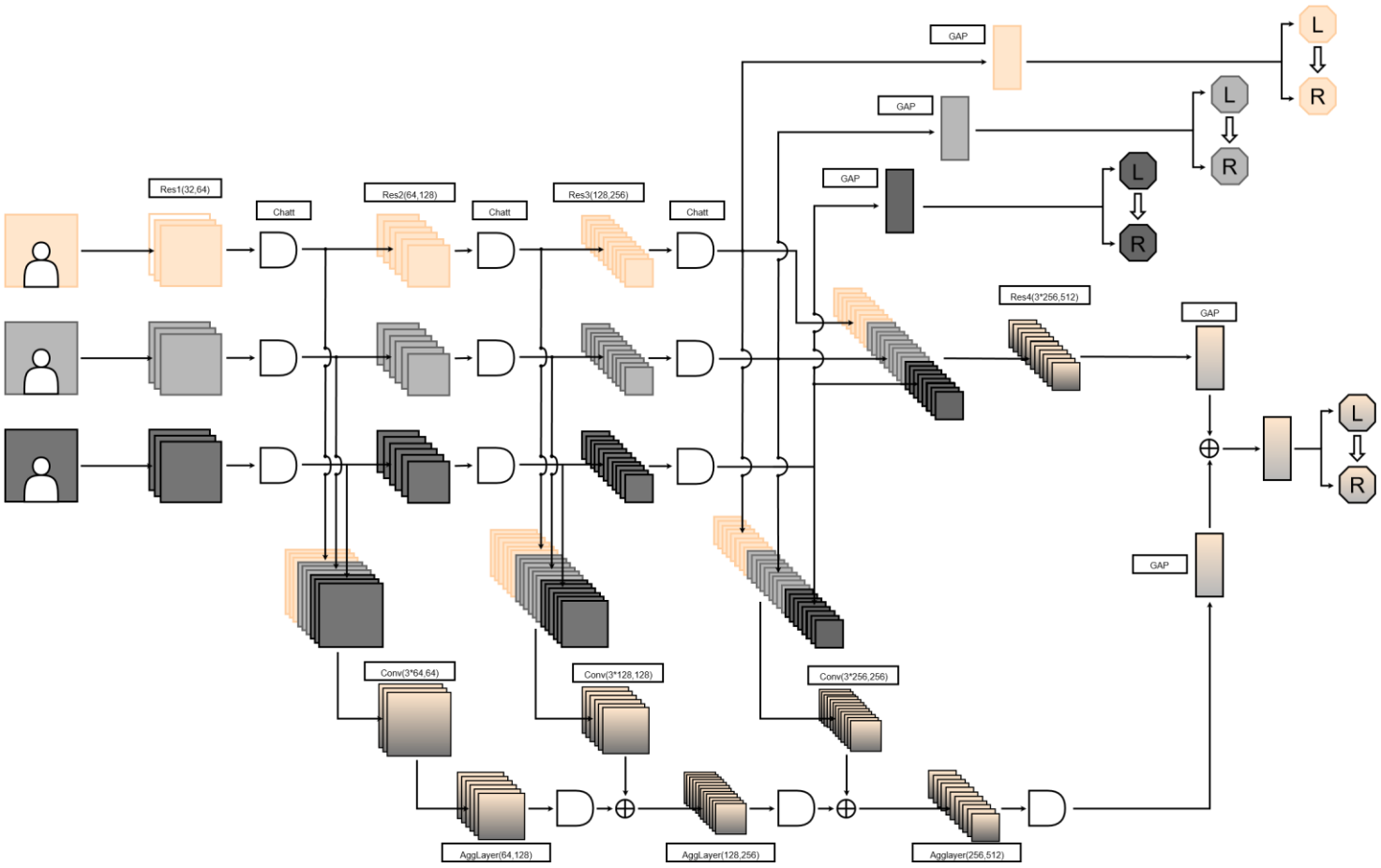

Figure 24 - Model architecture with all the components used

4.2.3. GUI

We created two GUIs using Python and QT Creator for image acquisition and for access control.

The image acquisition GUI (Figure 40) consists of buttons for creating a new subject, changing the subject, taking picture bursts and for changing the image type (if it is a real person or a spoof). We also have a scroll bar to see the RGB images in real-time. The GUI has 5 image displays, for each modality (RGB, IR, and Depth) and for the face type and illumination helper images. Finally we have a progress bar to count the image number for each modality in real-time.

To use this GUI the subject needs to sit in front of the Intel Realsense camera close enough that the image displayed in the RGB image display is cropped and segmented. Check which light or combination of lights need to be on or off and the image type and then press the “Take Pictures” button. The process of taking the pictures is then automatic, only allowing front faced

images. The images are saved in a buffer and are only saved to the disk after the complete burst. The images are saved to the disk in a thread to allow for a smooth visualization of the user displayed images.

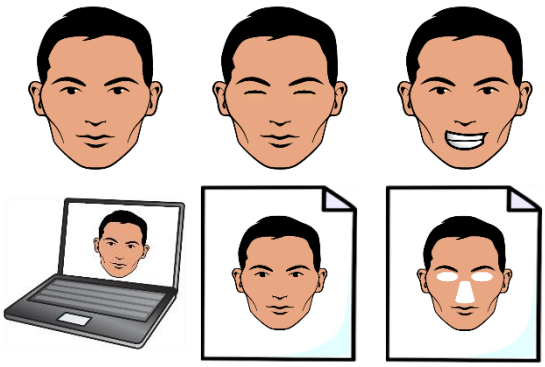

Figure 25 - Face type helper images (Top: Normal, Blinking, Talking/Smilling; Bottom: Screen attack, Print attack, Cut-out print attack)

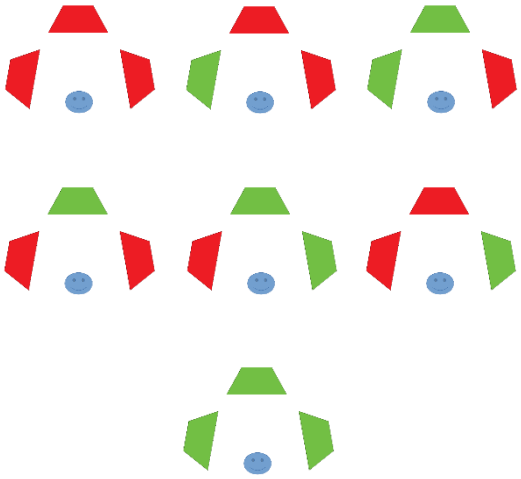

Figure 26 - Illumination helper images (Green: light on; Red: light off)

The access control GUI (Figure 41) has 4 different displays, one for each modality (RGB, IR, and Depth) and one for the local subject images that the network classifies the real-time subject, and a scrollbar for better visualization of said images.

This GUI is operated at the same time as our trained network. After loading the network's weights we use the captured real-time subject images to feed the network. We use our model's output to select the class with the highest probability and save them in a FIFO type buffer to display both liveness and recognition values. When the network detects a real person it displays the identity and the confidence it has that the subject is well classified as well as the liveness confidence and a green rectangle around the subject face. When the network decides the face is a spoof it displays "unknown" for the class and the confidence of the liveness detection as well as a red rectangle around the subject face.

4.2.4. Image Acquisition

4.2.4.1. Cameras Alignment

The RGB and Depth+Ir modules from the IntelRealsense D435 are not aligned, so we implement a standard alignment process using affine transformations.

We start by capturing an image for each modality at the same time. We manually select different points in the RGB image and repeat the process with the same points for the IR image. Using OpenCV function *cv2.estimateAffinePartial2D* we get the matrix transformation between both images that allows us to use OpenCV function *cv2.warpAffine* to align the RGB image to the IR image. In the end of this process we save the transformation matrix for future use.

Figure 27 - Top: Unaligned face images; Bottom: Aligned face images

4.2.4.2. Dataset Acquisition

In order to test our network in a real-world scenario, we had to create our face images dataset. We set the scene in the lab with multiple lighting and positions to obtain a more realistic view. For each anonymous subject, we take 900 pictures for each modality, totaling 2700 real pictures. We program a script to get bursts of 30 pictures from the person for different combinations of lighting and face variations (View Figure 29). We ask the subject to act normal for the real subject pictures, only moving and rotating their head slightly. We limit the rotation angle of the head, so the face is mainly centered with the possibility of some slight rotation.

This constraint is implemented with a simple pixel distance measurement between the landmark pixels on the side of the face and the nose (View Figure 28). When the head has a higher rotation the distances will have a greater difference, so we discard those images.

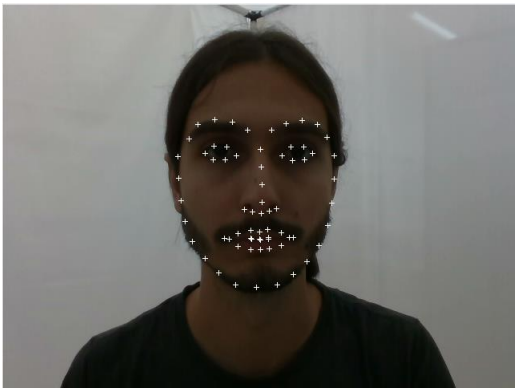

Figure 28 - Facial landmarks

For the spoof images, we took the pictures ourselves with some face cut-outs and with a computer (View Figure 30), similarly to the CASIA-Surf Dataset [5]. Below is a table explaining the details of the face images acquisition for each subject.

	Face Type	# Pictures for each Lighting type							Total
		N	L	C	R	LC	CR	LCR	
Real	Normal	90	30	30	30	30	30	60	300
	Closing/Opening eyes	90	30	30	30	30	30	60	300
	Talking	90	30	30	30	30	30	60	300
Spoof	Computer image attack	90	30	30	30	30	30	60	300
	Print image attack	90	30	30	30	30	30	60	300
	Cut-out print image attack	90	30	30	30	30	30	60	300
Total		540	180	180	180	180	180	360	1800

Label :

N - No Lighting
L - Left Lighting
C - Center Lighting
R - Right Lighting

Table 4 - Image acquisition details and number of images

Figure 29 - Various illuminations face images

Figure 30 - Spoof attacks from left to right: Computer image attack, Print attack, Cut-out Print attack

4.2.5. Image Preprocessing

4.2.5.1. Face detection and Segmentation

We can detect faces in the original images and obtain various landmarks using a pre-trained face detection network[9]**.** This network uses the *dlib* package for the face detection task, a basic *resne18* [49] for shape regression and is trained on a total of 4 datasets (AFW, HELEN, IBUG e LFPW).

Our network is designed to get segmented face images as input. We detect the faces in the original images obtained from the IntelRealsense camera using the *dlib* package for Python and pass the images through ISR CV Lab's face landmark network. We can then segment the original images with the resulting landmarks and perform a depth normalization obtaining the face image with the background set to zero, ready to pass through our liveness detection and facial recognition network.

The depth normalization converts the raw distances obtained from the camera to an image ranging from 0 to 255, with the pixel closest to the camera having the highest value. This process is only applied to the distances inside the masked area.

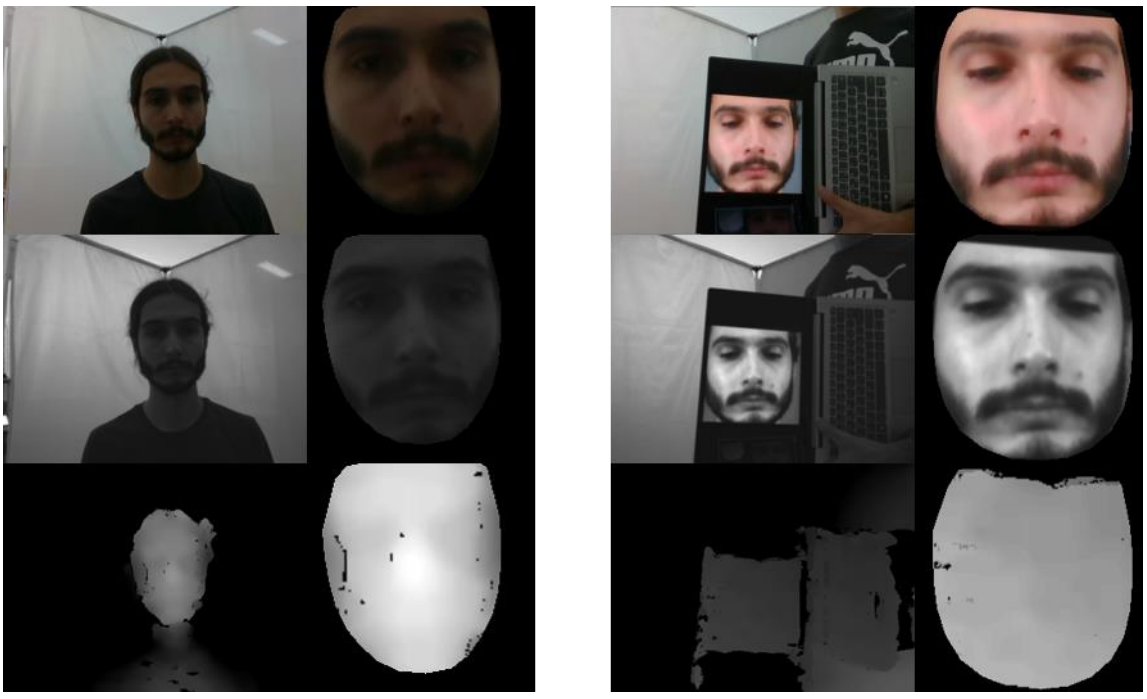

Figure 31 - Original and segmented images: real (left) and spoof (right) for each modality from top to bottom: RGB, IR and Depth

[9] Developed by the investigator Pedro Martins from the Instituto de Sistemas e Robotica (ISR) de Coimbra in the Computer Vision (CV) Lab

4.2.5.2. Data Augmentation

A useful method for improving the performance of any CNN is to use augmented data to avoid over-fitting the model. We implement a fake face shadow to mimic a real-world scene. We start by creating a binary mask the size of the face image. This mask has a random chance of drawing a circle or a rectangle with an arbitrary size, setting the values to 1's. With this mask, we can randomly change the intensity of the pixels, creating a virtual shadow that resembles the shadows created in the real world.

In addition to this augmentation, we also use some implementations of rotation, resizing, cut-out, horizontal flipping, grayscale, and normalization as performed by A. Parkin *et al.* [21].

Figure 32 - Input face image (left) and various fake shadow input face images (right)

4.2.6. Real time process

4.2.6.1. Getting images in real time

Using the realsense sdk[10] and pyrealsense[11] python module, we can obtain the three modalities images in real-time. The depth normalization provided by Realsense decodes the depth image with a dynamic function, resulting in a decoded face image that changes values if any object or person appears in the image, even if not directly in front of the face. With this in mind, we decide it is best to apply our normalization after cropping and segmenting the face

[10] https://www.intelrealsense.com/developers/
[11] https://pypi.org/project/pyrealsense/

in order to obtain a more realistic image that does not normalize according to the closest point to the camera but according to the depth itself. We set a maximum threshold distance of 1.3 meters to obtain a reasonably sized image in the end.

4.2.6.2. Real time pipeline

The final task we need to implement is the real-time access control process. With the implemented detection and segmentation network, we can acquire images in real-time, detect the faces in the images, segment them, and pass them through our network.

We perform a Softmax operation to the output of the network, which is an array of values, resulting in an array of probabilities. The index of the highest probability indicates the subject in the case of the recognition task and if it is a real person or fake in the case of the liveness detection task.

We implement a buffer to increase the robustness of the system, which holds both the highest probabilities of each task and the subject belonging to the recognition. A threshold value for acceptance of probability is set, allowing for better control of the buffer for the identity validation in real-time.

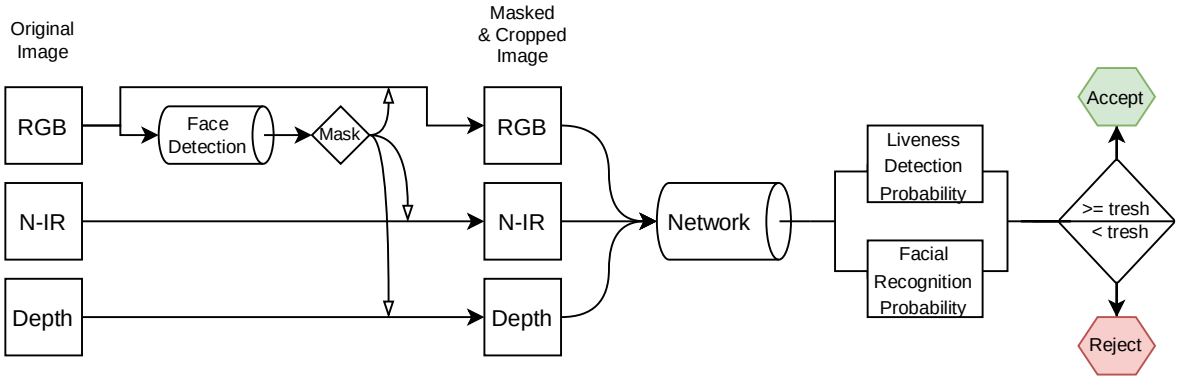

Figure 33 - Real-time pipeline

5

Results and Discussion

To evaluate the performance of our methods, we use the ROC curve as the primary evaluation metric for the liveness detection task and a typical accuracy metric for the recognition task.

A receiver operating characteristic curve (ROC) is a graphical plot that illustrates the binary classifier system as its differentiation threshold is varied. The ROC curve can be created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as recall or probability of detection and is calculated as $TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$. The false-positive rate is also known as the probability of false alarm and can be calculated as $FPR = \frac{FP}{N} = \frac{FP}{FP+TP}$. The ROC curve is a fitting indicator for the algorithms applied in real-world applications since we can select a proper threshold between FPR and TPR according to the requirements. We compute the ROC for TPR@FPR=10e−2, 10e−3, and 10e−4 as the quantitative indicators.

Since the ROC can only be used for binary classification systems, we use an average accuracy metric, which can be calculated as $ACC = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN}$, where TP is the number of True Positives, TN is the number of True Negative, FP is the number of False Positives and FN is the number of False Negatives.

For comparison with the ChaLearn Challenge [48] results, we provide the metrics ACER, APCER, NPCER, FAR and FRR in addition to previous metrics. We also provide the number of TP, TN, FP and FN for each experiment. The metrics are calculated as follows:

$APCER = \frac{FP}{TN + FP}$; $BPCER = \frac{FN}{TP + FN}$; $ACER = \frac{APCER + BPCER}{2}$; $FAR = \frac{FP}{FP+TN}$; $FRR = \frac{FN}{TP+FN}$

Following the introduction of the metrics, we present the evaluation of the CNN architectures for liveness detection and face recognition in our private dataset and the CASIA-Surf (previously mentioned in 2.5); and present the results in Table 5 and Table 6 respectively. We also experimented transfer learning approaches with both datasets presented in Table 7.

In our private dataset experiments, we trained the ResnetDLA architecture for both tasks, liveness detection (LD) and facial recognition (FR), with the classical Cross-Entropy Loss (ResnetDLA + CCE), A-Softmax Loss (ResnetDLA + SF), the novel Conditional A-Softmax Loss (ResnetDLA + CSF), the ResnetDLA with Channel Attention Modules (ResnetDLA + Chatt) and finally the ResnetDLA without the fake shadow data augmentation technique (ResnetDLA + NFS).

As we can see in Table 5, the Cross-Entropy Loss cannot differentiate each subject enough in the FR task, achieving an accuracy of 99.938%, while the A-Softmax Loss can achieve 100%. After replacing the A-Softmax Loss with our Conditional A-Softmax Loss we obtain the same results of 100% accuracy for both tasks with minor variations in the validation loss.

With the addition of the Channel Attention Module to the model architecture with our loss, we expected the model to learn better characteristics of the human face to improve the LD task but the results show that the model can't achieve the top accuracy of 100%, only reaching 99.938%, even though the recognition accuracy achieves top score as previous experiments.

Lastly, using the fake shadow data augmentation on the model with our loss allows for a more diverse dataset, especially in the spoof images, and the model also achieves 100% accuracy in both LD and FR tasks.

After analyzing the results obtained for our private dataset, we can conclude that the model architecture with our loss, either with or without the fake shadow data augmentation, achieves the best results of 100% accuracy for both tasks. These are remarkable results but considering our dataset is limited we cannot cover all the possible outcomes, and thus more experiments are required.

Motivated by the good results observed in Table 5 we repeated the train and evaluation of the model with the Conditional A-Softmax Loss in the extended CASIA-Surf Dataset. In order to improve the results even more we also tested the Multi-Branch Loss. The results confirm the improvement of the model with the use of the Multi-Branch Loss in comparison with the model with only Conditional A-Softmax Loss, improving the LD task by 0,013% and the FR task by 0,448%, achieving 100,00% and 99.591% in the respective tasks, which can be observed in Table 6. Even though both models reach essentially the same accuracy, the accuracy of the model with the multi-branch loss (Figure 35) achieves a peak in the very early epochs and is more stable, while the model without this method converges later (Figure 34). Since the final evaluation results for the LD task reach essentially a top 100% accuracy we can verify the ROC results in Figure 36 and Figure 37. We can also see that in the confusion matrix there is almost a straight line in the diagonal in both models (Figure 38, Figure 39) that indicates that the classes were mostly correctly identified.

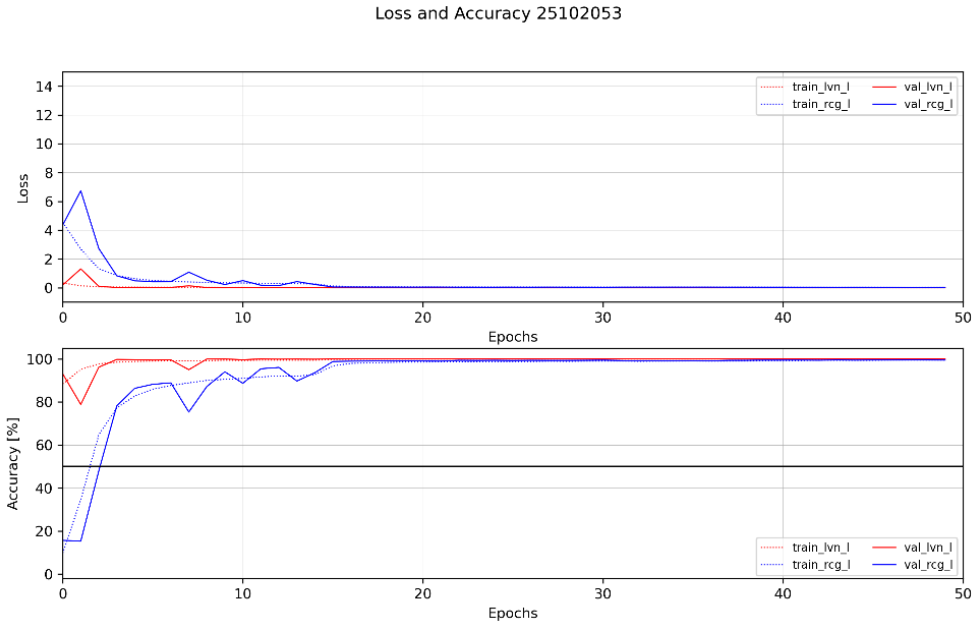

Figure 34 - Our Model Loss and Accuracy with ResNet18 backbone and A-Softmax Loss over 50 epochs in the mixed dataset

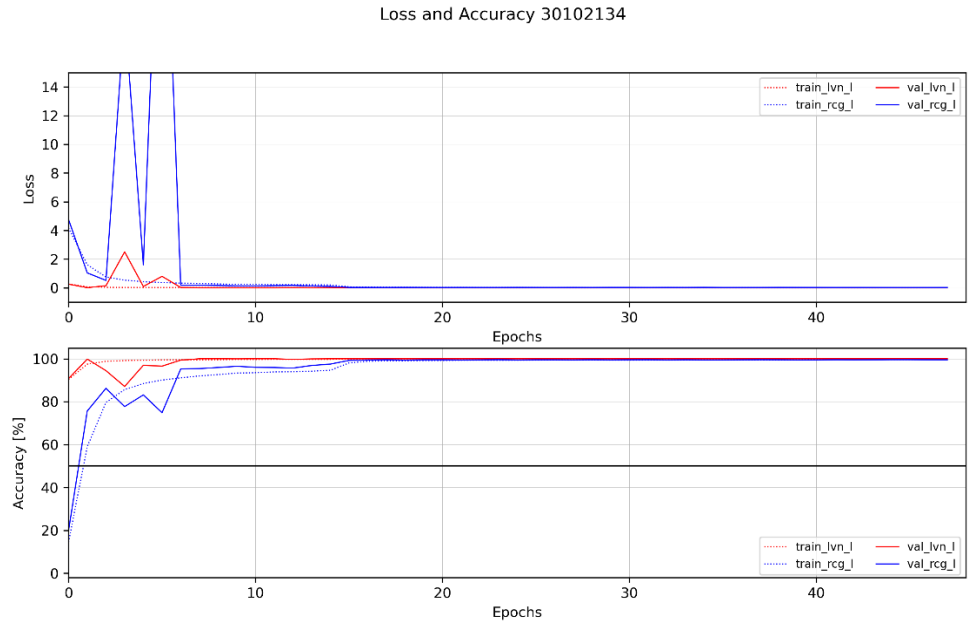

Figure 35 - Our Model Loss and Accuracy with ResNet18 backbone and multi-branch loss with A-Softmax Loss over 50 epochs in the mixed dataset

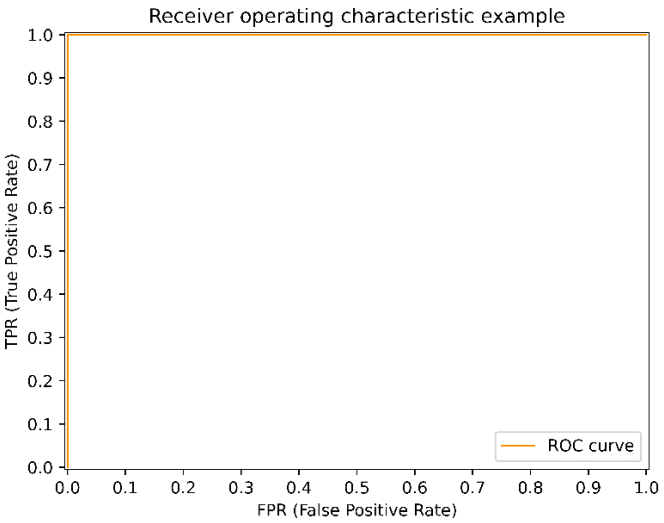

Figure 36 – ROC for our model with ResNet18 backbone and multi-branch loss with Conditional A-Softmax Loss for Liveness Detection in the mixed dataset

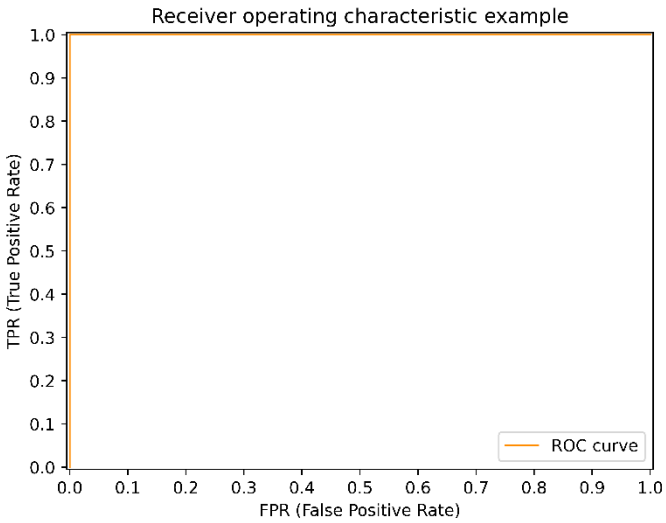

Figure 37 - ROC for our model with ResNet18 backbone with Conditional A-Softmax Loss for Liveness Detection in the mixed dataset

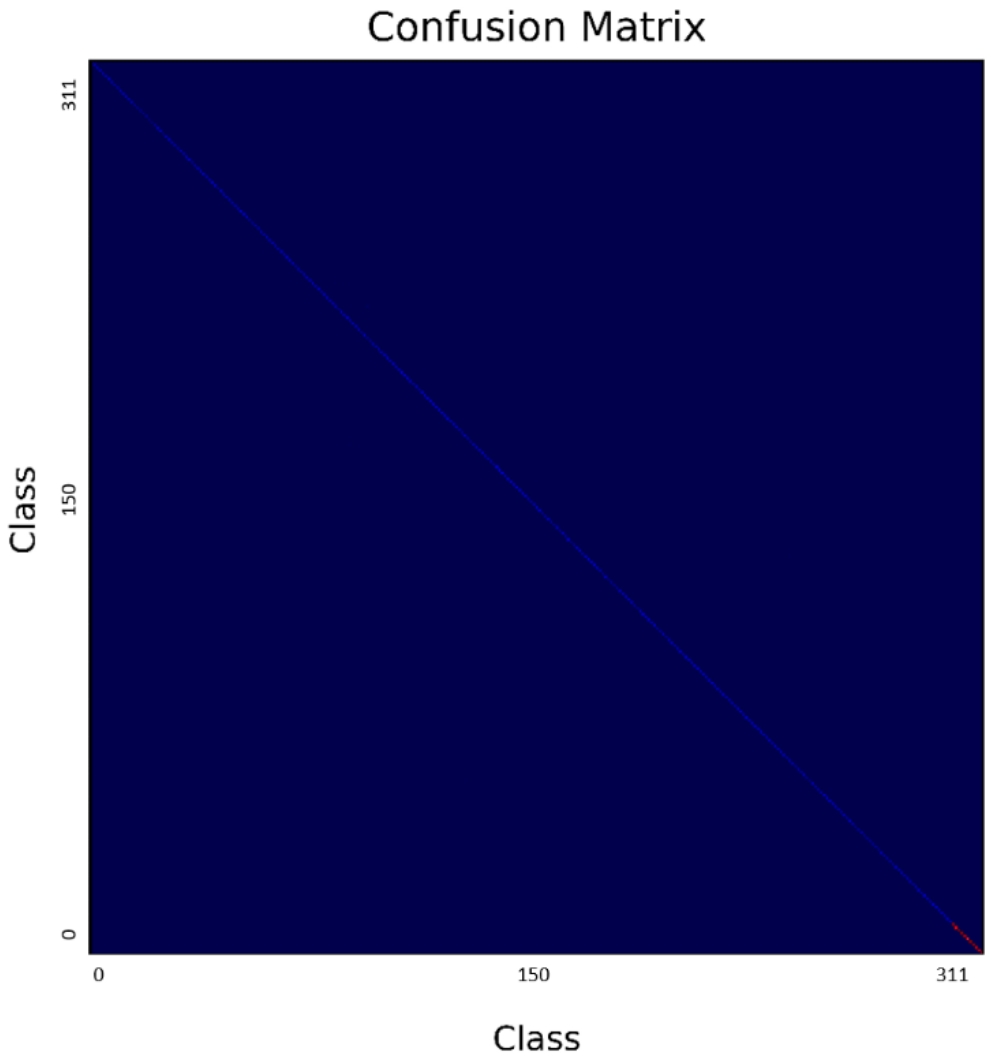

Figure 38 – Confusion Matrix for our model with ResNet18 backbone with A-Softmax Loss for Facial Recognition in the mixed dataset

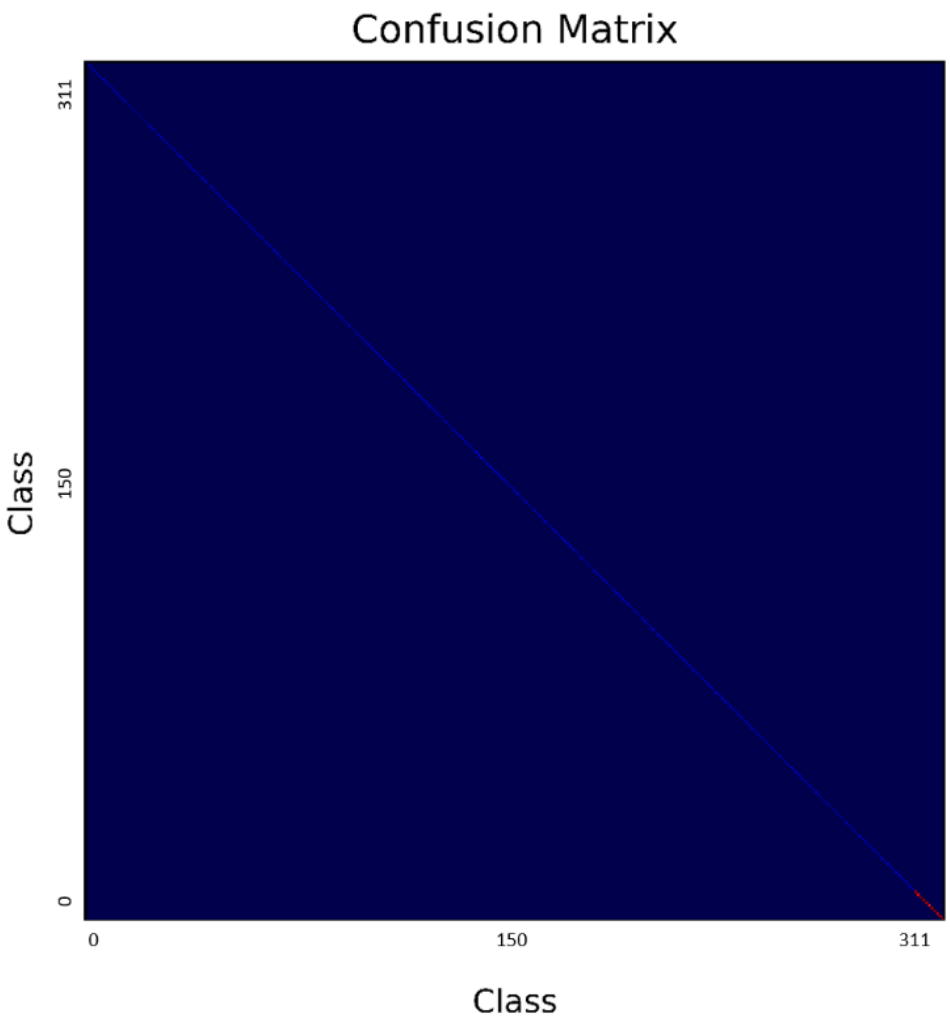

Figure 39 - Confusion Matrix for our model with ResNet18 backbone and multi-branch loss with A-Softmax Loss for Facial Recognition in the mixed dataset

Lastly, we tested our model with transfer learning for the liveness task, by training the network with the CASIA-Surf Dataset and then testing with both datasets: CASIA-Surf and our private dataset. We can verify from the results in Table 7 that the transfer learning approach is viable for this task since we can train the model in a large dataset and use the pre-trained model to perform the liveness detection task without the need to train the model every time a subject is added to the dataset.

Backbone + modules	Recognition		Liveness								
	Loss	Accuracy (%)	Loss	Accuracy (%)	TPR (%)			APCER (%)	NPCER (%)	ACER (%)	TP	FP	TN	FN
					@FPR 10e-2	@FPR 10e-3	@FPR 10e-4							
ResnetDLA + CCE	0. 001801	99.938	0. 000180	100.00	100.00	100.00	100.00	0.0	0.0	0.0	607	0	995	0
ResnetDLA + SF	0. 000120	100.00	0. 000135	100.00	100.00	100.00	100.00	0.0	0.0	0.0	607	0	995	0
ResnetDLA + CSF	0. 000242	100.00	0. 000055	100.00	100.00	100.00	100.00	0.0	0.0	0.0	607	0	995	0
ResnetDLA + CSF + NFS	0.000098	100.00	0.000057	100.00	100.00	100.00	100.00	0.0	0.0	0.0	607	0	995	0
ResnetDLA + CSF + CHATT	0.000259	100.00	0.000566	99.938	100.00	100.00	99.96	0.1005	0.0	0.0502	607	1	994	0

Label:

- CCE: Cross Entropy Loss
- SF: SphereFace Loss
- CSF: Conditional SphereFace Loss
- CHATT: Model with Channel Attention module
- NFS: No Fake Shadows data augmentation
- 3B: Multi-branch Loss

Table 5 - Training results of our model on our private dataset using all the modalities

Backbone + modules	Recognition		Liveness									
	Loss	Accuracy (%)	Loss	Accuracy (%)	TPR (%)			APCER (%)	NPCER (%)	ACER (%)	TP	FP	TN	FN
					@FPR 10e-2	@FPR 10e-3	@FPR 10e-4							
Resnet18 + CSF	0.035715	99.143	0.000710	99.987	100.00	100.00	99.98	0.0	0. 0396	0. 01984	2519	0	5061	1
Resnet18 + CSF + 3B	0.016626	99.591	0.000320	100.00	100.00	100.00	100.00	0.0	0.0	0.0	2520	0	5061	0

Table 6 - Training results of our model on the CASIA-SURF Dataset extended with our private dataset using all the modalities

Datasets		Liveness										
Train	Test	Loss	Accuracy (%)	TPR (%)			APCER (%)	NPCER (%)	ACER (%)	TP	FP	TN	FN
				@FPR 10e-2	@FPR 10e-3	@FPR 10e-4							
CASIA	CASIA	0.001624	99,916	100,00	99,94	99,84	0.0491	0.1568	0.1030	1910	2	4064	3
CASIA	Ours	0. 000345	100.00	100.00	100.00	100.00	0.0	0.0	0.0	607	0	995	0

Table 7 - Liveness Detection results with different training and testing datasets

6

Conclusions and Future Work

The goal of this dissertation was to develop a Liveness Detection and Facial Recognition system using multi-modal features. To achieve this objective we first based our network on a state-of-the-art anti-spoofing network and extended it to be able to perform facial recognition on the CASIA-SURF Dataset. After extensive testing on the CASIA-SURF we created our own dataset with images collected from a small group of participants in our laboratory.

After analyzing the results from all the experiments, it was possible to conclude that our system can achieve, in our dataset, a maximum accuracy result of 100.00% for the recognition task and TPR@FPR=10e-4 of 100.00% for the liveness task. The results are better than expected but more experiments need to be run to better evaluate our system.
As we know the channel attention plays an important part on the liveness detection and facial recognition task but in our tests we couldn't improve the results with this module, even though the results were only marginally worse than the tests we run without the module.
We can also conclude that even though the Cross Entropy Loss is widely used for the liveness detection task, this loss is not able to discriminate the classes with certainty for the facial recognition task. Even though the loss value is lower than the SphereFace Loss, the recognition accuracy is also lower, with a margin of 0.062% decrease in accuracy.

When we created our private dataset, we tried to create a well-balanced dataset with the same number of real images and fake images. Although the model validation presented good results, we noticed the system failed when tested as a real-time application outside the controlled environment. So we tried to change the balance of the dataset and achieved better performance. We concluded that the dataset needed to be unbalanced to avoid model overfitting, so the number of used authentic images for training was reduced, resulting in a proportion of $\frac{1}{3}$ real images to fake images.

Considering the work developed and the results obtained, there are several possibilities for continuing the project to improve the current work, such as:

- Implementation of a more robust and developed dataset with more subjects and spoof attack types, such as morphed face images or 3D masks;
- Implementation of other loss functions such as the LMCL;
- Implementation of new backbone models and modality fusion methods;
- Improvement of the face detection task;

References

[1] K. Lai, S. Samoil, and S. N. Yanushkevich, “Multi-spectral facial biometrics in access control,” *arXiv*, 2020.

[2] J. Guo, X. Zhu, C. Zhao, D. Cao, Z. Lei, and S. Z. Li, “Learning meta face recognition in unseen domains,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, no. September, pp. 6162–6172, 2020, doi: 10.1109/CVPR42600.2020.00620.

[3] Y. Zhang *et al.*, “CelebA-Spoof: Large-Scale Face Anti-spoofing Dataset with Rich Annotations,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12357 LNCS, pp. 70–85, 2020, doi: 10.1007/978-3-030-58610-2_5.

[4] Y. Liu, A. Jourabloo, and X. Liu, “Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 389–398, 2018, doi: 10.1109/CVPR.2018.00048.

[5] S. Zhang *et al.*, “CASIA-SURF: A Large-Scale Multi-Modal Benchmark for Face Anti-Spoofing,” *IEEE Trans. Biometrics, Behav. Identity Sci.*, vol. 2, no. 2, pp. 182–193, 2020, doi: 10.1109/tbiom.2020.2973001.

[6] A. Lagorio, M. Tistarelli, M. Cadoni, C. Fookes, and S. Sridharan, “Liveness detection based on 3D face shape analysis,” 2013, doi: 10.1109/IWBF.2013.6547310.

[7] L. Sun, W. Huang, and M. Wu, “TIR / VIS Correlation for Liveness Detection in,” pp. 114–121, 2011.

[8] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, “Computationally efficient face spoofing detection with motion magnification,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 105–110, 2013, doi: 10.1109/CVPRW.2013.23.

[9] J. Yan, Z. Zhang, Z. Lei, D. Yi, and S. Z. Li, “Face liveness detection by exploring multiple scenic clues,” *2012 12th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2012*, vol. 2012, no. December, pp. 188–193, 2012, doi: 10.1109/ICARCV.2012.6485156.

[10] S. Autherith and C. Pasquini, “Detecting Morphing Attacks through Face Geometry Features,” *J. Imaging*, vol. 6, no. 11, p. 115, 2020, doi: 10.3390/jimaging6110115.

[11] R. Raghavendra, K. B. Raja, and C. Busch, “Detecting morphed face images,” *2016 IEEE 8th Int. Conf. Biometrics Theory, Appl. Syst. BTAS 2016*, 2016, doi: 10.1109/BTAS.2016.7791169.

[12] I. Medvedev, F. Shadmand, L. Cruz, and N. Gonçalves, “Towards facial biometrics for ID document validation in mobile devices,” *Appl. Sci.*, vol. 11, no. 13, 2021, doi: 10.3390/app11136134.

[13] M. Ngan, P. Grother, K. Hanaoka, J. Kuo, and I. A. Division, “NISTIR 8292 DRAFT SUPPLEMENT Face Recognition Vendor Test (FRVT) Part 4 : MORPH - Performance of Automated Face Morph Detection NISTIR 8292 DRAFT SUPPLEMENT Face Recognition Vendor Test (FRVT) Part 4 : MORPH - Performance of Automated Face Morph Detec.”

[14] L. Feng *et al.*, “Integration of image quality and motion cues for face anti-spoofing: A neural network approach,” *J. Vis. Commun. Image Represent.*, vol. 38, pp. 451–460, 2016, doi: 10.1016/j.jvcir.2016.03.019.

[15] H. Chen, G. Hu, Z. Lei, Y. Chen, N. M. Robertson, and S. Z. Li, “Attention-Based Two-Stream Convolutional Networks for Face Spoofing Detection,” 2019.

[16] R. Shao, X. Lan, and P. C. Yuen, “Joint discriminative learning of deep dynamic textures for 3D mask face anti-spoofing,” *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 4, pp. 923–938, 2019, doi: 10.1109/TIFS.2018.2868230.

[17] Z. Yu *et al.*, “Multi-modal face anti-spoofing based on central difference networks,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, pp. 2766–2774, 2020, doi: 10.1109/CVPRW50498.2020.00333.

[18] T. Shen, Y. Huang, and Z. Tong, “FaceBagNet : Bag-of-local-features Model for Multi-modal Face Anti-spoofing,” no. 1.

[19] H. Kuang *et al.*, “Multi-modal multi-layer fusion network with average binary center loss for face anti-spoofing,” *MM 2019 - Proc. 27th ACM Int. Conf. Multimed.*, pp. 48–56, 2019, doi: 10.1145/3343031.3351001.
[20] S. Zhang *et al.*, “A dataset and benchmark for large-scale multi-modal face anti-spoofing,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 919–928, 2019, doi: 10.1109/CVPR.2019.00101.
[21] A. Parkin and O. Grinchuk, “Recognizing multi-modal face spoofing with face recognition networks,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2019-June, pp. 1617–1623, 2019, doi: 10.1109/CVPRW.2019.00204.
[22] C. C. Hsu, Y. X. Zhuang, and C. Y. Lee, “Deep fake image detection based on pairwise learning,” *Appl. Sci.*, vol. 10, no. 1, 2020, doi: 10.3390/app10010370.
[23] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, “Two-Stream Neural Networks for Tampered Face Detection,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1831–1839, 2017, doi: 10.1109/CVPRW.2017.229.
[24] Y. Kim, J. Na, S. Yoon, and J. Yi, “Masked fake face detection using radiance measurements,” vol. 26, no. 4, pp. 760–766, 2009.
[25] K. W. Bowyer, K. Chang, and P. Flynn, “A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition,” vol. 101, pp. 1–15, 2006, doi: 10.1016/j.cviu.2005.05.005.
[26] N. A. Spaun, “Facial comparisons by subject matter experts: Their role in biometrics and their training,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5558 LNCS, pp. 161–168, 2009, doi: 10.1007/978-3-642-01793-3_17.
[27] B. Heisele, P. Ho, J. Wu, and T. Poggio, “Face recognition: Component-based versus global approaches,” *Comput. Vis. Image Underst.*, vol. 91, no. 1–2, pp. 6–21, 2003, doi: 10.1016/S1077-3142(03)00073-0.
[28] P. S. Penev and J. J. Atick, “Local feature analysis: A general statistical theory for object representation,” *Netw. Comput. Neural Syst.*, vol. 7, no. 3, pp. 477–500, 1996, doi: 10.1088/0954-898X_7_3_002.
[29] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Face recognition using LDA-based algorithms,” *IEEE Trans. Neural Networks*, vol. 14, no. 1, pp. 195–200, 2003, doi: 10.1109/TNN.2002.806647.
[30] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997, doi: 10.1109/34.598228.
[31] R. Shyam and Y. N. Singh, “Evaluation of eigenfaces and fisherfaces using bray curtis dissimilarity metric,” *9th Int. Conf. Ind. Inf. Syst. ICIIS 2014*, pp. 0–5, 2015, doi: 10.1109/ICIINFS.2014.7036600.
[32] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, “Face recognition by independent component analysis,” *IEEE Trans. Neural Networks*, vol. 13, no. 6, pp. 1450–1464, 2002, doi: 10.1109/TNN.2002.804287.
[33] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on feature distributions,” *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996, doi: 10.1016/0031-3203(95)00067-4.
[34] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” *Adv. Neural Inf. Process. Syst.*, vol. 3, no. January, pp. 1988–1996, 2014.
[35] R. Shyam and Y. N. Singh, “Identifying Individuals using Multimodal Face Recognition Techniques,” *Procedia - Procedia Comput. Sci.*, vol. 48, no. Iccc, pp. 666–672, 2015, doi: 10.1016/j.procs.2015.04.150.
[36] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, “Targeting Ultimate Accuracy: Face Recognition via Deep Embedding,” pp. 1–5, 2015, [Online]. Available: http://arxiv.org/abs/1506.07310.
[37] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, “An All-In-One Convolutional Neural Network for Face Analysis,” *Proc. - 12th IEEE Int. Conf. Autom.*

Face Gesture Recognition, FG 2017 - 1st Int. Work. Adapt. Shot Learn. Gesture Underst. Prod. ASL4GUP 2017, Biometrics Wild, Bwild 2017, Heteroge, pp. 17–24, 2017, doi: 10.1109/FG.2017.137.

[38] C. Han, S. Shan, M. Kan, S. Wu, and X. Chen, "Face recognition with contrastive convolution," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11213 LNCS, pp. 120–135, 2018, doi: 10.1007/978-3-030-01240-3_8.

[39] J. Yang *et al.*, "Neural aggregation network for video face recognition," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5216–5225, 2017, doi: 10.1109/CVPR.2017.554.

[40] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep Learning for Face Anti-Spoofing: A Survey," pp. 1–25, 2021, [Online]. Available: http://arxiv.org/abs/2106.14948.

[41] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1701–1708, 2014, doi: 10.1109/CVPR.2014.220.

[42] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1891–1898, 2014, doi: 10.1109/CVPR.2014.244.

[43] Y. Wen, K. Zhang, Z. L. B, and Y. Qiao, "A Discriminative Feature Learning Approach," *Eccv*, vol. 1, pp. 499–515, 2016.

[44] H. Wang *et al.*, "CosFace: Large Margin Cosine Loss for Deep Face Recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 5265–5274, 2018, doi: 10.1109/CVPR.2018.00552.

[45] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6738–6746, 2017, doi: 10.1109/CVPR.2017.713.

[46] J.~S.~Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," *Adv. Neural Inf. Process. Syst.*, vol. 2, no. Ml, pp. 211–217, 1990.

[47] A. Liu *et al.*, "CASIA-SURF CeFA: A benchmark for multi-modal cross-ethnicity face anti-spoofing," *arXiv*, pp. 1–17, 2020.

[48] A. Liu *et al.*, "Multi-modal face anti-spoofing attack detection challenge at CVPR2019," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2019-June, pp. 1601–1610, 2019, doi: 10.1109/CVPRW.2019.00202.

[49] K. He, "Deep Residual Learning for Image Recognition."

[50] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," pp. 1–13.

[51] T. Agarwal, "PERFORMANCE COMPARISON OF DEEP," *2019 Twelfth Int. Conf. Contemp. Comput.*, pp. 1–6, 2019.

[52] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11211 LNCS, pp. 3–19, 2018, doi: 10.1007/978-3-030-01234-2_1.

Attachments

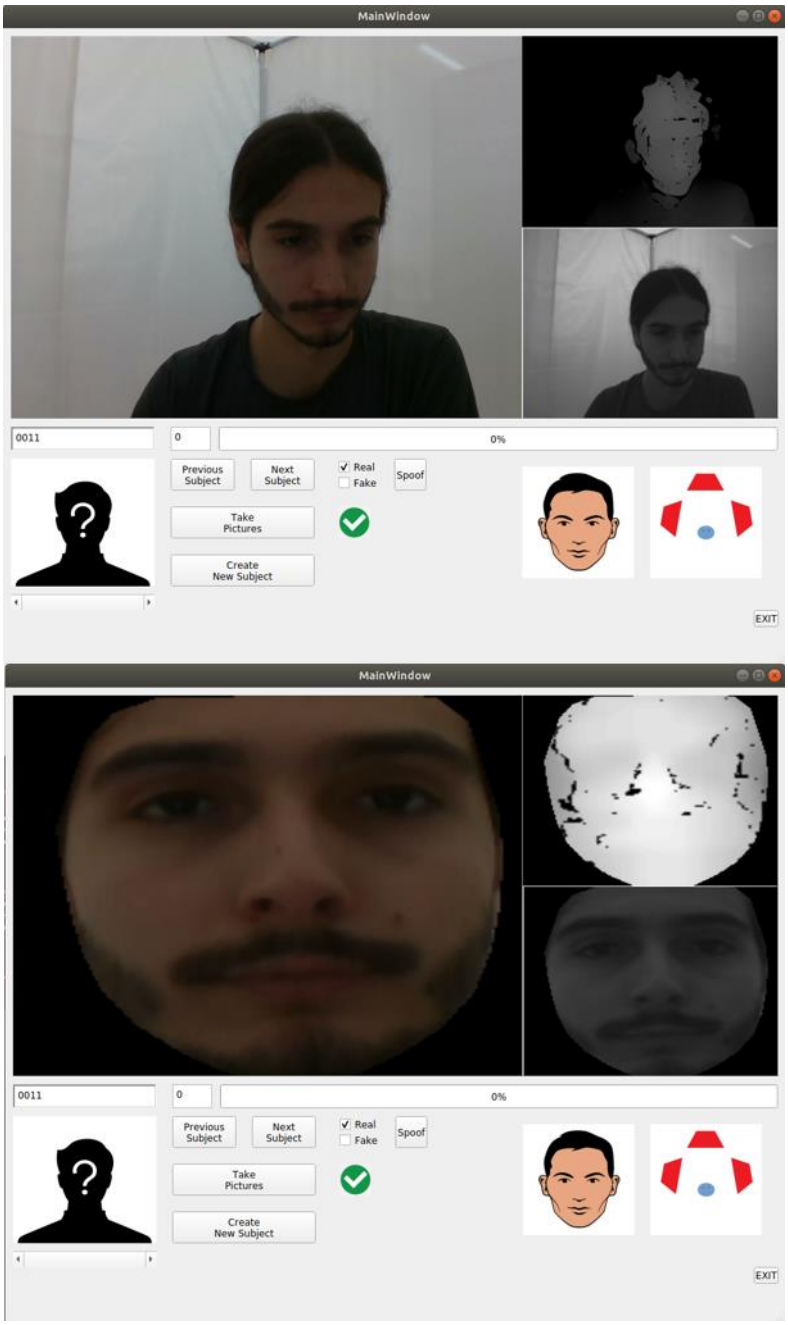

Figure 40 - Images acquisition GUI

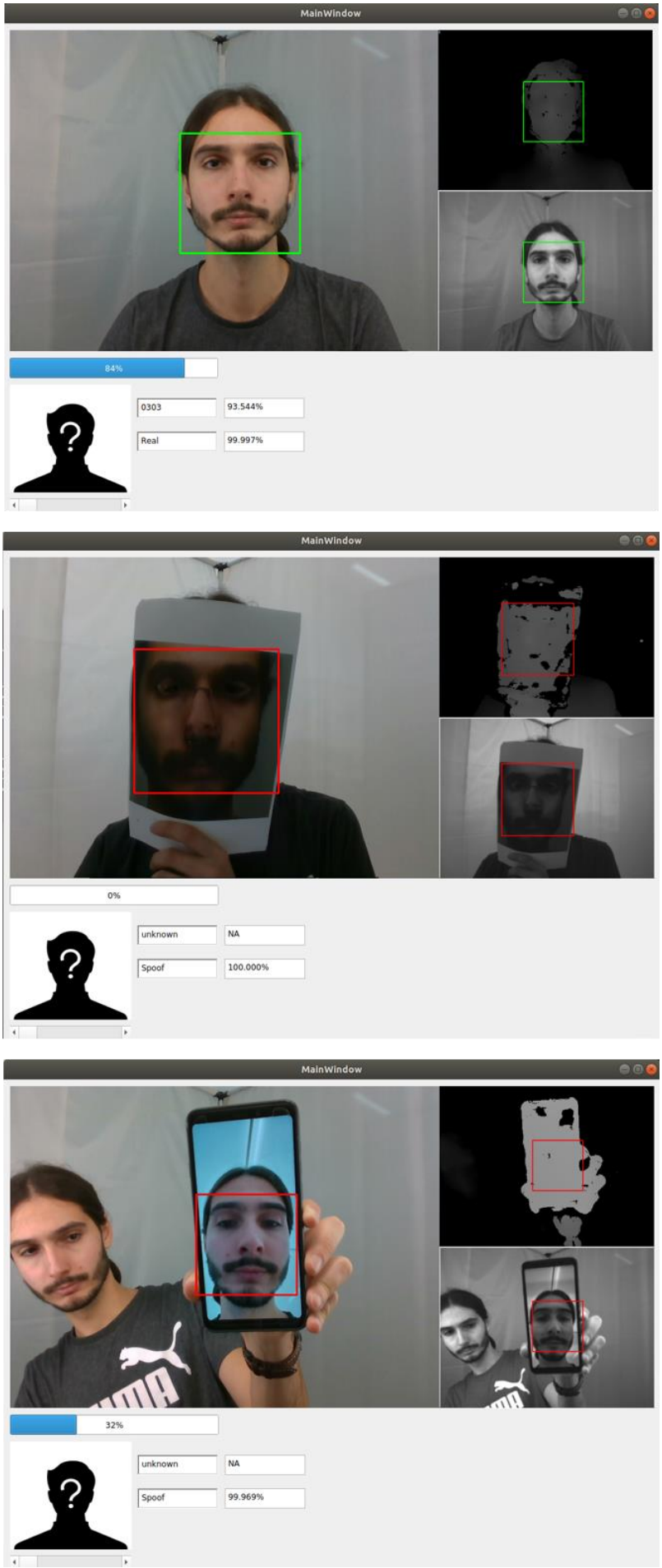

Figure 41 - Liveness Detection and Facial Recognition Real-Time GUI