UNIVERSIDADE D
COIMBRA

Miguel Nuno Leão Gordilho Hipólito Correia

Dealing with Overfitting in the Context of Liveness Detection using FeatherNets with RGB images

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores no Ramo de Computadores orientada pelo Professor Doutor Nuno Miguel Mendonça da Silva Gonçalves e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Setembro de 2022

UNIVERSIDADE D
COIMBRA

Miguel Nuno Leão Gordilho Hipólito Correia

Dealing with Overfitting in the Context of Liveness Detection using FeatherNets with RGB images

Dissertation carried out within the scope of the Integrated Master in Electrical and Computers Engineering, Computers branch, under the guidance of Professor Nuno Miguel Mendonça da Silva Gonçalves, presented to the Department of Electrical and Computers Engineering of the Faculty of Sciences and Technologies of the University of Coimbra

Jury:
Professor Doutor Luís Alberto da Silva Cruz
Professora Doutora Maria do Carmo Raposo de Medeiros
Professor Doutor Nuno Miguel Mendonça da Silva Gonçalves

Coimbra, 2022

This project was developed in collaboration with:

Abstract

Facial Anti Spoofing (FAS) or liveness detection, has gained a large interest with the increasing use of facial recognition in day-to-day activities and its requirement for security. From the variety of different approaches that have been developed, the use of machine learning solutions has become the more popular approach due to the improvement of these types of solutions for other problems as well as the increased number of available datasets for liveness detection. These models however carry shortcomings like overfitting, where the model adapts perfectly to the training set, becoming unusable when used with the testing set, defeating the purpose of machine learning. This thesis focuses on how to approach overfitting without altering the model used by focusing on the input and output information of the model.

The input approach focuses on the information obtained from the different modalities present in the datasets used, as well as how varied the information of these datasets is, not only in number of spoof types but as the ambient conditions when the videos were captured. The output approaches were focused on both the loss function, which has an effect on the actual ”learning” of the machine learning, used on the model which is calculated from the model’s output and is then propagated backwards, and the interpretation of said output to define what predictions are considered as bonafide or spoof. Throughout this work, the authors were able to reduce the overfitting effect with a difference between the best epoch and the average of the last fifty epochs from 36.57% to 3.63%.

Keywords: bonafide, spoof, overfitting, dataset, model.

List of Figures

List of Tables

Acronym List

ACER Average Classification Error Rate

ANN Artificial Neural Network

APCER Attack Presentation Classification Error Rate

BPCER Bonafide Presentation Classification Error Rate

CE Cross Entropy

CNN Convolutional Neural Network

EER Equal Error Rate

FAR False Acceptance Rate

FAS Face Anti-Spoofing

FL Focal Loss

FLOPS Floating Point Operations per Second

FN False Negative

FP False Positive

FRR False Rejection Rate

GAP Global Average Pooling

HOG Histogram of Gradients

HSV Hue Saturation Value

HTER Half Total Error Rate

LBP Local Binary Pattern

PA Presentation Attacks

PAD Presentation Attack Detection

PR Precision Recall

RGB Red Green Blue

rPPG Remote Photoplethysmography

SGD Stochastic Gradient Descent

TN True Negative

TNR True Negative Rate

TP True Positive

TPR True Positive Rate

YCbCr Luminosity (Y) Blue difference chroma (Cb) Red difference chroma (Cr)

ZSFA Zero Shot Face Anti-Spoofing

Contents

Chapter 1

Introduction

With the rise of facial recognition technology in day-to-day applications, such as mobile payments, comes a concern for the security of these systems. To counteract these security vulnerabilities, which present themselves as Presentation Attacks (PA), the development of Presentation Attack Detection (PAD), also known as Face Anti Spoofing (FAS) or liveness detection began. A PA or spoof can take on a large number of forms, be it the impersonation of another individual using methods as simple as a paper print of the victim's face or as complex as silicone masks, or the obfuscation of the attacker's identity. With the growing complexity of these attacks, so too the complexity of the PAD methods seems to grow in order to keep up.

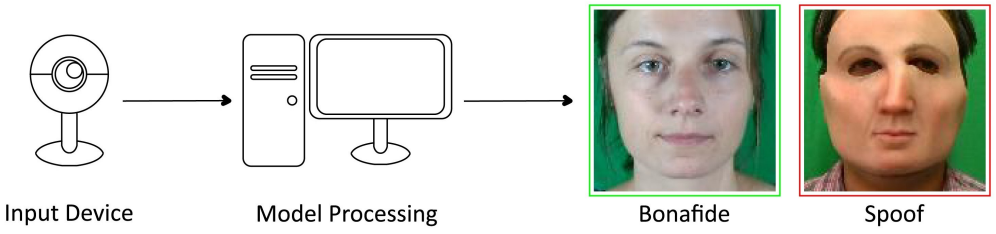


Figure 1.1: Liveness detection concept image. The approaches used attempt to recognize if the face that is presented is bonafide or not. Example images taken from [1].

As said, these systems are put in place along side facial recognition systems and in many aspects are very similar. Unlike facial detection, which searches for one or multiple faces, both facial recognition and liveness detection assume that the face has already been detected and is currently being presented to the system. Now while facial recognition searches through a database for the presented face, liveness detection does not search through a database for an individual but for signs of an attack. As such, liveness detection

can be employed before, after or simultaneously with facial recognition, if used before the facial recognition, while more efficient since if an attack is detected there is no need to search through the database, if the individual presents themselves truthfully and once the liveness detection check has been cleared, then present the attack in order to be recognized in the database, the system has been deceived. A more secure option is then to employ liveness detection after the face has been recognized in the system, thus guaranteeing security in both steps. The system may also try to guarantee that the image used for both operations is the same, ensuring that the subject remains the same during the process.

Liveness detection can be employed through active or passive systems. Active systems request action from the user to prove they are who they claim to be, this can come through request of certain motions like smiling or blinking their eyes. Of more interest to this thesis and the community at large, passive systems require no action from the presented individual, analyzing the image for features that could indicate a spoof.

1.1 Context

Currently, most methods are based in deep learning, more specifically Convolutional Neural Networks (CNN) or a variation of these, which are trained by feeding them large quantities of information extracted from datasets with images from various modalities, be it simple RGB images, depth maps or even infrared images. While most models use RGB images, these are almost always supplemented by one or more extra forms of information in order to help in achieving greater results. This information is obtained using high quality devices, the preparation of the data is time costly, due to both the quality and quantity of the data requires large amount of storage and the models require a lot of development time and computation cost to achieve a viable state.

The datasets consist in images/videos of various individuals presenting themselves as themselves, or as another individual being these cases considered bonafide and spoof respectively. The complexity of a dataset grows with the variety of attacks that it presents. As an explanation, consider a dataset with only bonafide cases that can have 10, 100 or 1000 individuals, between each one of them there are already many differences since no two people are alike in every aspect, now have each of them present themselves as another person present in the dataset. Depending on how many attacks you wish to include in your dataset, and assuming you want each person/spoof combination to be present, you add number of spoofs $\times$ number of individuals examples. Despite the increased requirements, the more representations present in a dataset, the more desirable it is since it bridges more and more the gap between controlled academic/industrial setting and real life.

These spoofs can be used for several purposes, but the interest of the community is

not in what they are used for but to be able to recognize them. The objective is to the apply said knowledge to systems which grant access to one thing or another with facial identification, granting them a new degree of security by being able to distinguish between legitimate and illegitimate accesses.

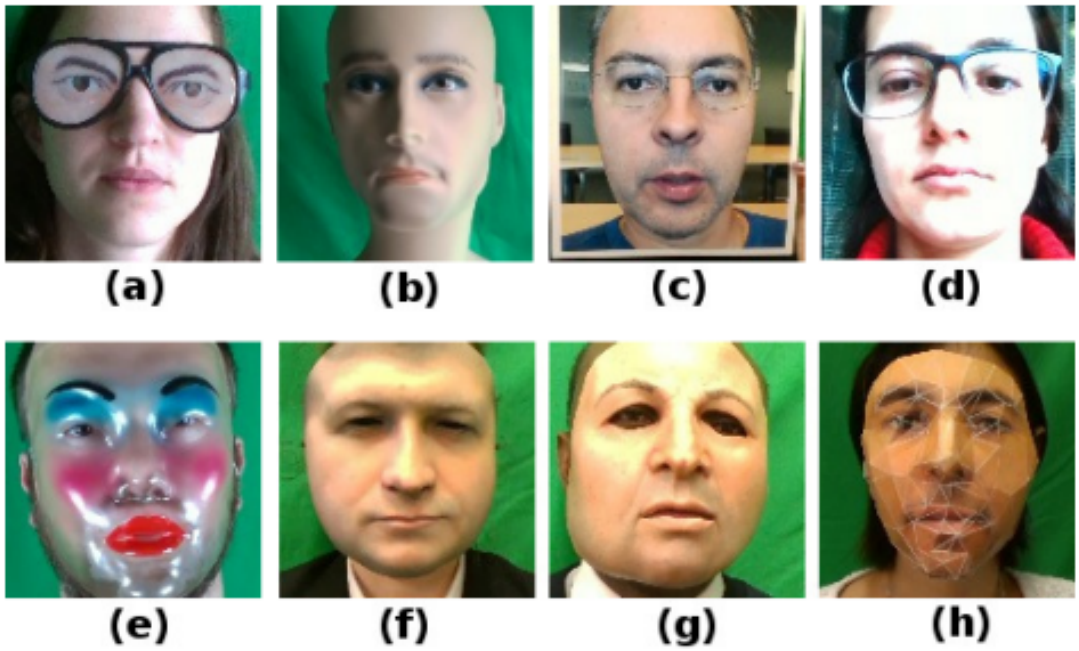


Figure 1.2: Different presentation attack examples. These are (a): glasses (funny eyes glasses), (b): fake head, (c): print, (d): replay, (e): rigid mask (decorative plastic mask), (f): rigid mask (custom made realistic), (g): flexible mask (custom made realistic), and (h): paper mask. With some more believable than others, there is an argument to be made on how certain attack types like ”funny glasses” and ”decorative masks” hardly qualify as impersonation and are better qualified as obfuscation attacks, and as such have little interest in capturing the resemblance of any individual. Taken from the documentation of [1].

At the same time, these systems that would most benefit from liveness detection, such as cellphones or web cameras, that can be used for entry to a platform using facial recognition, don’t have the capability of using the more successful i.e. more costly liveness detection approaches be it for a lack of camera quality, processing power, storage capacity or pure monetary cost of implementation.

1.2 Motivation

There are then two options so that this application can be possible: either the systems develop rapidly as to accommodate the more developed liveness detection approaches or

these approaches get simpler while maintaining their success rate. Of course, while easier said than done, the simpler of the two is the second option and this is what the community has continuously attempted since its inception, with many developments in reducing the cost of liveness detection approaching the problem at various parts of the models, applying the knowledge that has been accumulated from other issues that employ machine learning solutions, most commonly from other object recognition problems.

A common problem in machine learning is overfitting, where the model adapts itself too closely to a certain portion of data or to the noise present in the input, which then makes it unreliable to the full picture to which it was presented. Many of the complexities presented in FAS models are added precisely to avoid overfitting and the search for lighter and faster performance will always come at the cost of overall accuracy. This thesis attempts to explore alternatives that require little to no adaptation of the machine learning model itself, focusing on what data is fed to the model and how it is interpreted.

Beyond overfitting, there is the question of generalization which is how well a developed solution adapts to different scenarios from the one that was used for its development. This can be considered one of the end goals of not only liveness detection but any problem that employs machine learning for its solution. Though not the focus of this thesis, some of the conclusions drawn during this work may be taken into consideration to benefit the progress made for the development of generalized models.

1.3 Contributions

This thesis was able to take steps in studying and, eventually mitigating the effects of overfitting in machine learning solutions for liveness detection problems. The initial baseline result of 99.32% accuracy, obtained with depth images, gave little room for improvement so a new baseline using RGB images instead was obtained. These results are not only less successful with an accuracy of 89.75% , but display overfitting with the average of the model's last 50 accuracy results being 53.18%. Through this work, while unable to improve the result from the best epoch, the developed approaches were able to remove or at least heavily lower the overfitting effect, with the top accuracy of 89.37% then achieving an average of the final 50 results equal to 85.75%.

1.4 Document Structure

This document is divided in six chapters starting with the introduction that was just presented, followed by:

- Chapter 2 - State of the Art: this chapter will give a brief history of the multiple

developments that liveness detection has received, starting with the datasets created for the task, followed by the different approaches be they hand crafted, machine learning or somewhere in between;

- Chapter 3 - Background: this chapter is more focused in explaining more core concepts that are required to understand the proposed methodology;
- Chapter 4 - Methodology: this chapter presents everything that was used during the development of this thesis, and the details of how the experiments were conducted;
- Chapter 5 - Results and Discussion: here the results obtained from the experiments detailed in the previous chapter are presented and discussed;
- Chapter 6 - Conclusion and Future Work: finally the conclusions obtained during this thesis are presented and followed by future endeavours that can follow this work.

Chapter 2

State of the Art

2.1 Previous Approaches

There have been several approaches to deal with liveness detection that, as mentioned in the introduction, have improved and adapted to the more complex attacks that have been developed alongside them. These approaches can be divided into active and passive liveness detection, with active liveness detection requiring interaction from the user and passive liveness detection only requiring the images captured of the individual's face.

Active liveness detection won't be heavily discussed in this thesis with the focus staying on passive methods. However it is still interesting to consider the active approaches of liveness detection that used techniques such as tracking eye movement on screen (usually at request of the program) [2, 3], through facial expressions [4] or by the simple act of blinking [5]. It is important to note that what classifies these methods as active is the request, either direct or indirect, made to the user to follow a set of instructions, as these methods could be employed in passive approaches. These methods, while effective, become obsolete if the attacker is able to perform the requested tasks while maintaining the spoofed identity. These approaches are considered further in the text when the attacks are presented.

To overcome these shortcomings, **passive liveness detection** approaches make no requests of the user and instead analyze the image for signs of spoofing. This can be achieved through liveness cues, much like the active approaches, going as searching for an heart pulse using remote photoplethysmography (rPPG) [6] but these of course maintain the same shortcomings as previously stated. The interest is then to find patterns in the spoof images through descriptors such as HOG [7] and LBP [8], calling them (and also technically all the previously mentioned methods) as handcrafted methods.

Opposed to handcrafted approaches, where a programmer defines the conditions that signal when a spoof occurs, there are machine learning approaches, where a developed network will learn what is bonafide and what is spoofed. The transition to machine learning came from the large development that the area received, and since it proved great for object or pattern recognition in other fields, its use for liveness detection was only logical. Since the question of liveness detection can be put bluntly as ”bonafide or spoof” the first machine learning solutions used employ binary cross-entropy loss as the sole learning supervision for the network [9, 10, 11], however due to its simplicity, the models are prone to overfitting since they can easily focus their learning in arbitrary features, not relevant to the liveness detection problem. While the use of different loss functions [12, 13] by interpreting the issue in other ways, another solution was to aid the loss function using pixel-wise supervision.

Pixel-wise supervision can be made by using previous knowledge of liveness detection, and applying it to the model. For example, the use of pseudo depth maps [14, 15] based on the knowledge that, two dimensional attacks (print and replay) will display a ”flat” depth map can be used to aid the model. With this information, Atoum et al. [16] created ”DepthNet” capable of using these depth maps as evidence used for the model. By the same logic, binary mask labels [17, 18] or reflection maps [19] have been used.

The previously mentioned approaches are all based on color inputs (RGB, YCbCr or HSV) and it is the modality most commonly used. However, thanks to the development in sensors, it is possible to retrieve datasets using other modalities like depth, infra-red or thermal images. The models can then use a singular type of modality, or use the information available from several modalities all at once, fusing them at the input level like Nikisins et al. [20] that joined the information provided by the different modalities to then conclude that more information may be gained from isolated face patches rather than the whole face. The fusion can then be made at the feature level like Huafeng et al. [21] that lift the features obtained at various points in the model’s structure, obtained from different modalities and then fuse them together to then correctly adjust the decision weights. Finally, the fusion can be made at the decision level like Zhang et al. [22] in FeatherNets **(the model used for this thesis)** that uses results obtained from different modalities and other models to help with more ambiguous decisions.

There are some works that try to address the problem of generalization through the development of the model. Due to several factors like illumination, camera quality, facial position, types of attack, etc... models tend to be unable to maintain their success when tested in different conditions to their initial ones. The problem may be tackled through domain adaptation where the model tries to bridge the gap between the source and target domains (training and testing set) by allowing the model to map a function that relates the

two sets [23]. Another approach is domain generalization, where the objective is to focus the learning of the model only in essential features of the domains considered, for example Kim et al. [24] found that by efficiently suppressing irrelevant factors like illumination, the generalization capabilities of their model were improved.

The development of the methods presented doesn't necessarily always start from the bottom, with many of the networks developed by the community being used as a backbone for future works. As an example, MobileNets [25, 26, 27] which were developed for use in mobile and embedded video applications, and optimised through each version, with version 3 being tuned specifically to mobile phone's CPUs. The models were built to be both efficient and malleable, so that developers could adjust the network's hyper-parameters to their specific problem, with the interest of applying liveness detection to applications employed on mobile and embedded devices. Residual Networks (ResNets) [28] that are based on the idea that shallower (less layers) networks are easier to train than deeper ones, also serve as the backbone of several works, as well as the previously mention DepthNet are both commonly used as backbones for liveness detection models [29, 30], [31, 32, 33]. Finally it can be mentioned that these networks may be used indirectly, like in the case of FeatherNets which is inspired by both the architecture of MobileNetsV2 and the focus of creating lightweight networks able to function with systems of lower processing power.

2.2 Datasets

Datasets are an essential part of any machine learning development, varying in data type, size and quality among other attributes and variables. While facial recognition/detection has been in development since the 1960's [34] and as such has accumulated a large number of datasets, the interest in liveness detection only began in the 2010's [35], however in this short time span, various datasets have been developed by researchers and the industry, steadily increasing the number of individuals present, the number of images/videos, the quality and image modalities, and perhaps of most interest to this dissertation the number of different attacks present. Currently these attacks, considering their variations, are:

Print Attacks which are the presentation of a photograph to a facial recognition program. These are easily accessible due to the amount of face photos in both official documents as well as in unofficial platforms such as social networks. One variation is the quality of the print which is affected not only by the quality of the image itself but also from the type of paper used for the print or the quality of the printer. These images can then also receive cutouts of certain key features of the face such as the eyes, nose or mouth so when the picture is placed over the face of the attacker, it sits flush against it;

Mannequin Attacks are a more difficult to succeed type of attack since, the skill required to correctly capture an individuals semblance through sculpting, far surpassed the skill required to obtain an individuals photograph. This in conjunction with these attacks having the same shortcomings as print attacks against active liveness detection systems, not being able to respond to the presented commands, would make mannequin attacks a non issue since they would not be regularly used since the high fabrication requirements don't translate into an high amount of successful attacks. However the fact that the attack is three dimensional brings with it advantages against liveness detection solutions based on depth information;

Figure 2.1: Example of a print attack. Taken from [36].

Replay Attacks which are similar to print attacks, with the presentation being made using a digital screen instead of a paper print. This opens the possibility of presenting a video recording of the individual instead of a still image, which is capable of tricking the active liveness detection approaches if these are previously known, by requesting the spoofed individual to perform them beforehand. These videos wouldn't be viably attainable in a real life situation but the attacks can be considered none the less since through social engineering or, less subtly, through coercion, both the videos and the conditions of any active liveness detection system could be obtained. Again, similar to print attacks, these attacks can vary through the type of camera that is used and the type of screen that presents the attack, the higher the quality of botch capture and presentation, the more viable the attack;

Figure 2.2: Example of a replay attack. Taken from [36].

Mask Attacks are the most complex attack type currently available in datasets. While there are mask attacks that use novelty masks, like some of the examples shown in figure 1.2, the more custom made silicone masks either flexible or rigid, are very successful in capturing the semblance of individuals. They are also effective when used against active liveness detection, since the person wearing the mask is able to follow the commands given without knowing them previously, and against passive liveness detection models that are reliant in depth or infra red information since the attack is not only presented in three dimensions, but the heat radiating from the presented individual is, depending on the material, not affected by the mask.

The previously mentioned attacks are all impersonation attacks, that comprise the large majority of information present in datasets. However, some datasets [17, 38] present some obfuscation attacks like makeup, tattoos, glasses or wigs, but these types of attacks are not very prevalent for the community.

Figure 2.3: Samples of the CASIA-SURF 3DMask dataset. These are examples of the more true to life type of mask attacks, being able to capture the semblance of the bonafide individual more accurately. The left four columns are indoor while the right two ones are outdoor scenes. Taken from [37].

From table 2.1 some notable cases can be pointed out like Replay-Attack being one of the oldest datasets available thus being frequently used while there didn't exist many options but is recently used in conjunction with CASIA-MFSD for cross-dataset testing in order to evaluate the generalization capabilities of a model. On the topic of generalization, the OULU-NPU dataset, whose acquisition was made using 6 different smartphone cameras and its attacks (print and replay) came from 2 different printers and 2 different display devices, throughout 3 different sessions each with their own lighting, is divided in 4 different protocols used to evaluate the generalization capabilities of a model in different areas:

- Protocol I evaluates the capabilities under different environmental situations, namely differences in illumination and the background scenario;

- Protocol II evaluates the capabilities under different display devices both for the print attacks and replay attacks;
- Protocol III evaluates the capabilities under different acquisition devices i.e., the different cameras used to acquire the videos of the individuals;
- Protocol IV is the most challenging one by combining the previous protocols and testing the method with the result.

There are also interesting cases like SiW-M which was created with Zero-Shot Face Anti-spoofing (ZSFA) in mind. The concept of ZSFA is the ability to detect a previously unknown spoof, a problem that arises from the inability of researchers to predict what the next attack that will be invented is. This is achieved with the large variety of spoofs present, training the model to be able to detect any attack that is presented to it, even ones not present at its training. To show an adaptation of already existing datasets to the liveness detection problem, there is CelebA-Spoof, built upon the CelebA dataset [70] which is comprised of images of various celebrities, Zhang et al. [65] then used these images as their bonafide cases and created an assortment of spoofs from them, annotating each image with 43 attributes (gender, hair colour, expression, etc...). The spoofs are further characterized by the spoof type, varying the type of display for the replay attacks and the positioning of the print for the print attacks, and by the illumination and background present when the spoof was captured.

Finally, the two datasets used for this thesis are CASIA-SURF and WMCA, which are further detailed in section 4.

Dataset & Reference	Year	#Bonafide/Spoof	#Individuals	Modalities	#Spoof Types
NUAA [39]	2010	5,105/7,509 (I)	15	VIS	2
YALE-Recaptured [40]	2011	640/1,920 (I)	10	VIS	1
CASIA-MFSD [36]	2012	150/450 (V)	50	VIS	4
Replay-Attack [41]	2012	200/1,000 (V)	50	VIS	3
Kose and Dugelay [42]	2013	200/198 (I)	20	VIS	1
3DMAD [43]	2013	170/85 (V)	17	VIS, Depth	3
MSU-MFSD [44]	2014	70/210 (V)	35	VIS	3
UVAD [45]	2015	808/16,268 (V)	404	VIS	1
GUC-LiFFAD [46]	2015	1,798/3,028 (V)	80	Light Field	3
Replay-Mobile [47]	2016	390/640 (V)	40	VIS	2
HKBU-MARs V2 [48]	2016	504/504 (V)	12	VIS	2
MSU USSA [49]	2016	1,140/9,120 (I)	1,140	VIS	4
3DFS-DB [50]	2016	260/260 (V)	26	VIS, Depth	1
BRSU [51]	2016	102/404 (I)	137	VIS, SWIR	4
Msspoof [52]	2016	1,470/3,024 (I)	21	VIS, NIR	1
SMAD [53]	2017	65/65 (V)	-	VIS	1
OULU-NPU [54]	2017	720/2,880 (V)	55	VIS	2
MLFP [55]	2017	150/1,200 (V)	10	VIS, NIR, Thermal	2
ERPA [56]	2017	Total 86 (V)	5	VIS, Depth, NIR, Thermal	3
Rose-Youtu [57]	2018	500/2,850 (V)	20	VIS	5
SiW [58]	2018	1,320/3,300 (V)	165	VIS	5
LF-SAD [59]	2018	328/596 (I)	50	Light Field	3
CSMAD [60]	2018	104/159 (I&V)	14	VIS, Depth, NIR, Thermal	1
WFFD [61]	2019	2,440/2,445 (I&V)	745	VIS	1
SiW-M [17]	2019	660/968 (V)	493	VIS	13
3DMA [62]	2019	536/384 (V)	67	VIS, NIR	1
CASIA-SURF [63]	**2019**	**3,000/18,000 (V)**	**1,000**	**VIS, Depth, NIR**	**3**
WMCA [1]	**2019**	**347/1,332 (V)**	**72**	**VIS, Depth, NIR, Thermal**	**7**
Swax [64]	2020	Total 1922 (I&V)	55	VIS	1
CelebA-Spoof [65]	2020	156,384/469,153 (I)	10,177	VIS	7
RECOD-Mtablet [66]	2020	450/1,800 (V)	45	VIS	2
CASIA-SURF 3DMask [37]	2020	288/864 (V)	48	VIS	1
CeFA [67]	2020	6,300/27,900 (V)	1,607	VIS, Depth, NIR	5
HQ-WMCA [38]	2020	555/2,349 (V)	51	VIS, Depth, NIR, SWIR, Thermal	12
HifiMask [68]	2021	13,650/40,950 (V)	75	VIS	3
PADISI-Face [69]	2021	1,105/924 (V)	360	VIS, Depth, NIR, SWIR, Thermal	9

Table 2.1: List of available datasets. For the modalities presented, VIS states color images be them RGB or other, NIR stands for Near Infra Red and SWIR stands for Short Wave Infra Red. The I's and V's next to the number of cases stands for images or videos respectively. Information sourced from [35]2

Chapter 3

Background

3.1 Architecture

The initial approaches to liveness detection were based on searching for features inherent to spoofed images by using typical computer vision techniques like descriptors. Despite the effectiveness of these methods, the evolution made in machine learning approaches and their success in object classification or pattern recognition tasks made the jump to using them for liveness detection, an easy choice.

Artificial Neural Networks (ANN) are machine learning systems heavily inspired on how the human nervous system functions, comprised of a high number of interconnected computational nodes who try to collectively learn from the input to optimise the final output [71]. Convolutional Neural Networks (CNN) are a sub-class of ANNs being primarily used in pattern recognition in images [72] (a perfect fit for liveness detection). A generic CNN is comprised of four parts:

1. The input layer will hold the pixel values of the image;
2. The convolutional layer as the name indicates is essential to the CNN; it is here that the convolution occurs based on learnable kernels. These kernels are small in dimension but work through the entire input calculating a value, from these values the network will learn the kernels that react to a specific feature. While it is possible to train an ANN with image input, these cases aren't effective due to the fact that all neurons are connected, CNN's neurons are only connected to a small region of the input, having the fully-connected layer to connect everything;
3. The pooling layer aims to lower the required computation complexity of the model by scaling down the map obtained from the previous layer, usually with max-pooling. Max-pooling usually consists in picking the biggest value in 2x2 kernels with a

stride of 2, this scales down the convolution layer's output to 25% but preserves its depth;

4. The fully-connected layers will then produce scores from the previous layers activation's to be used for classification;

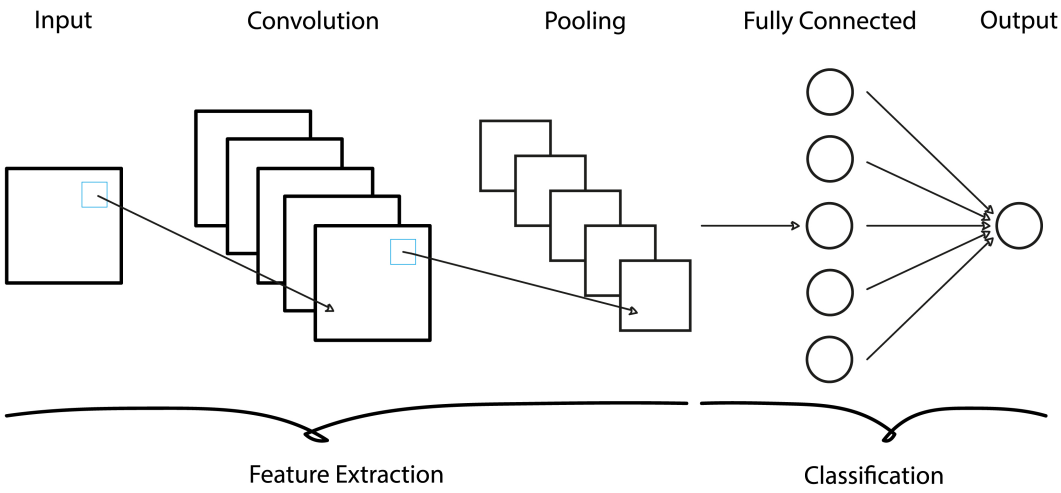


Figure 3.1: Schematic diagram of a basic CNN architecture.

3.2 Overfitting

Overfitting is an issue that affects many if not all machine learning solutions, not just those used for liveness detection problems. The problem arises when the model perfectly adapts to the training data used, making it unable to accurately make predictions when presented with the unseen testing data, defeating the purpose of the machine learning solution [73].

The previous picture displays how overfitting is shown according to the present data. However with the large amount of data present in the datasets used for liveness detection, it is not viable to search for overfitting by graphing the function to the data. A simpler approach is to analyze how overfitting affects the final results, with the losses decreasing until zero along the training set and the losses for the test set decreasing as well until a certain point where they begin to increase again, resulting in a "U-shaped" curve displayed

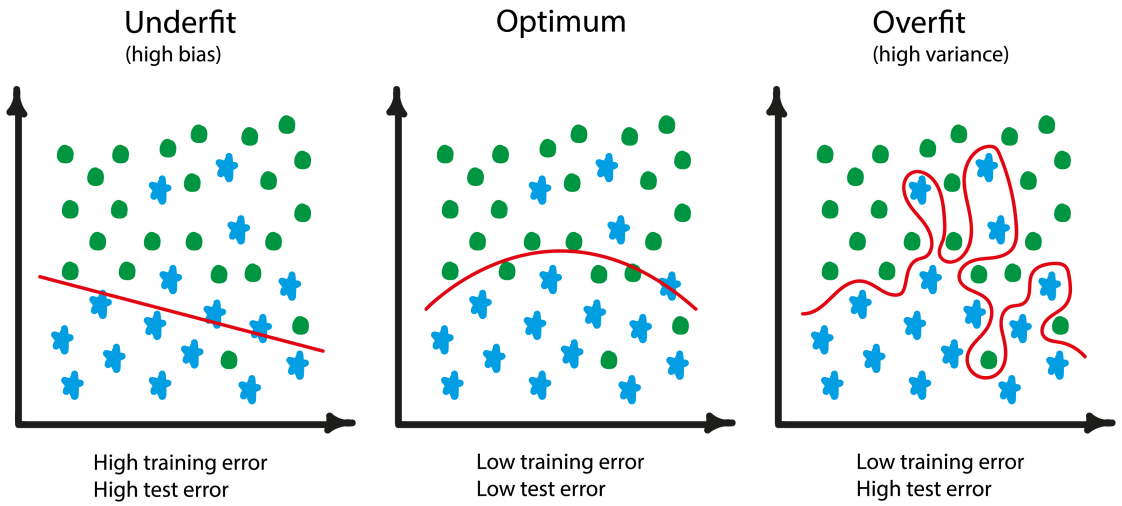


Figure 3.2: Visualization of underfitting versus overfitting. As the model (represented as the red line) adapts further to a certain set of data, the success towards the overall data may decrease.

in figure 3.3 (a). This curve then is inverted when analysing the accuracy graphs, lower loss translates to higher accuracy. Beside the graphical approach, the difference between the best results and the final ones can show how prevalent the overfitting effect is (the bigger the difference, the deeper the ”U”).

There are several approaches one may take to counter overfitting with none of them being completely effective since, when used in real life, the amount of hypothesis of data variation presented to the model is too great for it to properly work. The more common approaches are:

1. **Early stopping** - The more immediate approach is to stop the learning process before the model completely adapts to the training data. This requires at least one execution of the model during a large number of epochs as to find the ”sweet-spot” that corresponds to the lowest testing error;

2. **Network reduction** - Since the model will eventually adapt to the noise present in the data, a possible approach is to reduce the amount of learning that can be made with the noise information. This can be achieved by either placing criteria that stops adding conditions to the model’s rule or by dividing the training data into two sets,

one for learning and one for reduction with the second part of the training set now affecting the conditions made by the first part of the set through how relevant its information is;

3. **Expansion of the training data** - Closing the gap between the training set and testing set can substantially reduce overfitting since when the model perfectly adapts to the training set, it can still apply to the test data. Including more information by increasing the dataset to include more, ideally equal, hypotheses for both parts of the dataset, requires more labor hours in both capturing the information as well as labeling it. A ”cheaper” approach is data augmentation through the manipulation of already existing data is the more common approach;

4. **Regularization** - The model's output is affected by all features presented to it, the more features the more complex the model. These features can be more or less useful yet affect the model in the same way. The idea is to then either remove the useless features or at least reduce the weight they have in the model.

Recently, researchers found that complex models keep learning beyond zero loss for the training set, called ”interpolation” of the training set, and go on to present very low testing set error [74]. This goes against most literature on the topic and is explained with a double curve presented in figure 3.3 which is achieved by increasing the number of features or the size of the network's architecture.

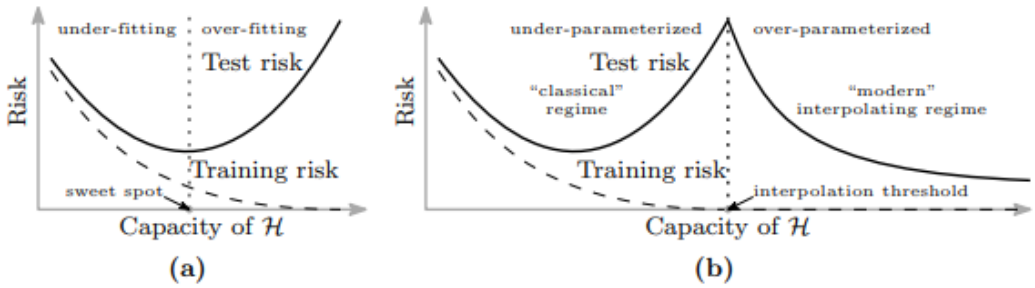


Figure 3.3: Curves for training risk (dashed line) and test risk (solid line). H stands for any one function class. (a) The classical U-shaped risk curve arising from the bias-variance trade-off. (b) The double descent risk curve, which incorporates the U-shaped risk curve (i.e., the “classical” regime) together with the observed behavior from using high capacity function classes (i.e., the “modern” interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk. Taken from [74].

3.3 Confidence

It is important to detail how we interpret confidence in a binary problem such as liveness detection, as opposed to confidence in multi label problems like object recognition. When presented with a yes or no question, the model doesn't answer ”yes” or ”no”, in reality the model will always answer ”yes”, only it does so with a certain degree of confidence, it is then up to the user to interpret this answer as they see fit.

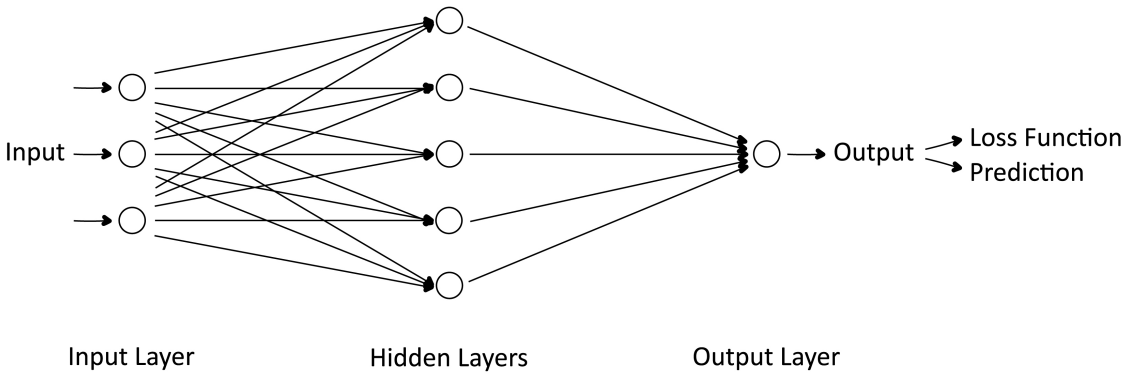


Figure 3.4: Schematic diagram of a basic machine learning architecture. The final output of the model is what is considered as confidence for this thesis. From the confidence score two things are calculated: the loss function, which then propagates backwards thus affecting the learning of the model, and the prediction which is simply stored to calculate the final confusion matrix, showing the distribution of predicted vs real labels.

This confidence score can be obtained in several forms according to how the model is implemented, however the more common and perhaps most simple to interpret is a score between 0 and 1 that translates to a percentage. In the binary problem, one can then define that anything above a 50% confidence score can be interpreted as ”yes” and anything below that can be interpreted as ”no”, but in reality, you wouldn't be very satisfied with ”fifty fifty” odds of someone successfully passing as yourself and gaining access to perhaps your bank account. As such this threshold value is something of interest and requires further study.

3.4 Loss Functions

In the context of deep learning, a loss function is what evaluates how successfully the model is performing: the lower the losses, the higher the success. Janocha and Czarnecki

state that most of deep learning models use binary cross entropy loss also known as log loss [75]. This applies well to liveness detection, considering that the problem is at its root a simple yes or no problem: ”Is this face bonafide or not?”. However, due to the simplicity of the loss function, these models can easily learn arbitrary patterns that deviate from the initial question of bonafide vs. spoof.

There have been several approaches attempting to solve the shortcomings of binary cross entropy loss mostly by using pixel-wise supervision which, as the name suggests, adds a new metric to the formula of the typical binary loss function by weighing in the effect of new information to the decision making. This method has proven to be a successful approach to solving both overfitting and erroneous decision making in the training process. One such example are pseudo depth maps [76, 77] which, despite being successful, require an additional labelling effort that in the end isn't as effective when presented with attacks that have actual three dimensions e.g. masks, mannequins.

A simpler and more effective approach is the pixel-wise supervision using binary mask labelling, which by isolating the facial region lowers the impact of arbitrary factors in the training of the network. George and Marcel [78] are the first to suggest this approach, with additions made over time in attempt to fine tune the original method.

3.5 Evaluation Metrics

In order to measure the success of any proposed method in Face Anti-Spoofing, there is a number of metrics that can be taken from the result's confusion matrix. The confusion matrix itself is a grid between the expected result and what was achieved, in binary cases like the basic approach to liveness detection, one can immediately take the values for the true positive (TP), false positive (FP), true negative (TN) and false negative (FN) with which the following metrics can be calculated [79]:

- Accuracy: The percentage of correct predictions on the dataset;

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.1}$$

- Recall: Also known as True Positive Rate (TPR) is the percentage of true values predicted as such;

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

- Specificity: Also known as True Negative Rate (TNR) is the percentage of false

values predicted as such;

$$Specificity = \frac{TN}{TN + FP} \tag{3.3}$$

- Precision: The percentage of correctly predicted true cases among all predicted true cases;

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

- False Acceptance Rate: The percentage of false cases that are wrongly accepted as true cases;

$$FAR = \frac{FP}{FP + TN} = 1 - Specificity \tag{3.5}$$

- False Rejection Rate: The percentage of true cases that are wrongly mistaken for false cases;

$$FRR = \frac{FN}{FN + TP} = 1 - Recall \tag{3.6}$$

- Half Total Error Rate: The average of the previous two metrics;

$$HTER = \frac{FAR + FRR}{2} \tag{3.7}$$

- Equal Error Rate: EER is the HTER when FAR and FRR are equal;

Recently the terms Attack Presentation Classification Error Rate (APCER), Bonafide Presentation Classification Error Rate (BPCER) and Average Classification Error Rate (ACER) have been used to evaluate liveness detection solutions. Simply put, APCER is equivalent to FAR measuring the amount of spoof cases that are considered as bonafide, BPCER to FRR measuring the amount of bonafide cases considered as spoofs and ACER to HTER being the average of the two.

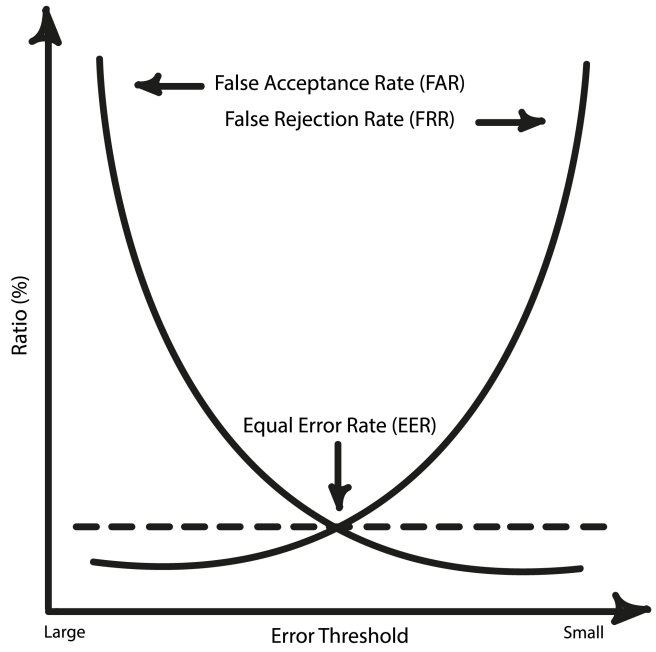


Figure 3.5: Relation between EER, FRR and FAR.

Chapter 4

Methodology

4.1 Datasets

The selection for the datasets used in this thesis came from the network used. For their work in FeatherNets, Zhang et al. [22] use CASIA-SURF [63]. Being a multi modal dataset, it has both the depth information used in the original work, as well as the color information used in this thesis. The objective was then to find a dataset in similar conditions that had more variety specifically in the number of spoof types. For this, the WMCA [1] dataset was perfect since both were captured using an Intel RealSense 3000 camera, and WMCA presents several new spoof types. Both datasets are detailed in the following sections.

4.1.1 CASIA-SURF

Developed by Zhang et al. [63] CASIA-SURF presents a larger dataset than most with 21,000 videos of 1,000 individuals captured with an Intel Real Sense 3000 camera providing not only RGB images but also depth and infrared images. The information is neatly distributed with one bonafide video to six spoof videos of each individual in each of the modalities provided by the camera. However where the dataset might be considered lacking is in the number of different attacks, the six spoof videos are all of print attacks. The print attacks were diversified by how the print was placed over the individuals face: either flat or pressed curved, and also the features of the print that were cut off: first removing the eyes, then the nose and finally the mouth. The conditions in which the videos were captured in a fixed setup where the individual stands in front of a green screen which displays various backgrounds without specified changes to the lighting, the individuals were then requested to tilt their heads, move closer and further away from the camera and move up and down.

	Training	Validation	Testing	Total
# Individuals	300	100	600	1000
# Videos	6,300	2,100	12,600	21,000
# Frames	1,563,919	501,886	3,109,985	5,175,790
# Sampled Frames	151,635	49,770	302,559	503,964
# Cropped Frames	148,089	48,789	295,664	492,552

Table 4.1: Statistical information of the CASIA-SURF dataset. Aside from the 300, 100 and 600 distribution of individuals for training, validation and testing sets respectively. From the large selection of frames, those that aren't viable for face recognition are removed. Information sourced from [63]

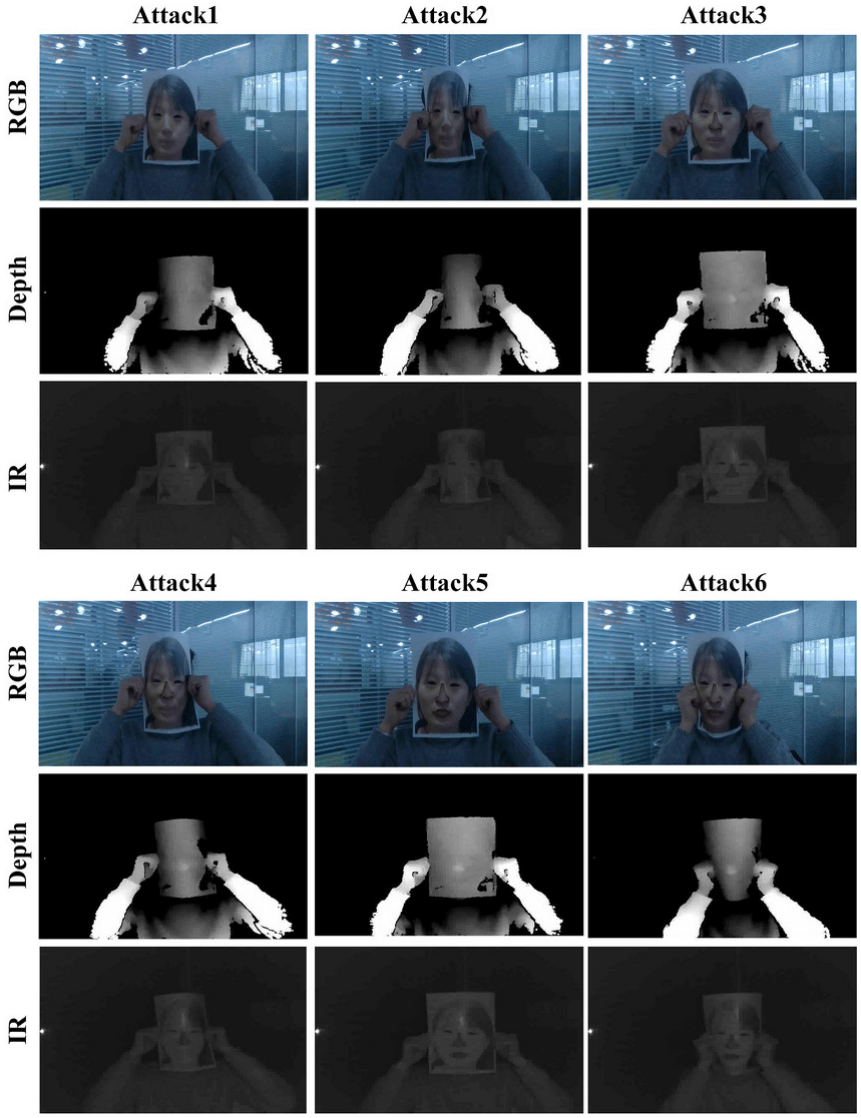


Figure 4.1: Attack examples of CASIA-SURF. Taken from [63]

CASIA-SURF's use was a simple choice due to the fact that it is one of the datasets used by Zhang et al. [22] in the development of their FeatherNets. During their devel-

opment, CASIA-SURF was improved with the addition of a Multi-Modal Face Dataset (MMFD) consisting of 43,853 videos of 15 subjects being divided in 15.415 bonafide cases and 28,438 spoofs. MMFD is currently a private dataset and as such cannot be accessed, this is not a prevalent issue since in terms of attack types it does not add any new ones to CASIA-SURF only expanding on the described variations its print attacks.

4.1.2 WMCA

Developed by George et al. [1] WMCA is quite smaller then the previous dataset with 1,679 videos of 72 individuals divided in 347 bonafide cases and 1,332 spoofs. This dataset was constructed with the same camera as CASIA-SURF having the same modalities, yet they added a Seek Thermal Compact PRO to capture thermal imagery of the individuals. Despite the lower number of videos, WMCA has the advantage of having a larger variety of attacks than CASIA-SURF adding to the print attacks, video replays, glasses, fake heads (mannequins), rigid masks, flexible masks and paper masks. These videos were captured with the individual on a fixed position through seven different sessions, in these sessions both the background and lighting varying, through uniform and complex backgrounds and through natural light, ceiling lighting and LED lighting.

	Training	**Validation**	**Testing**	**Total**
# Spoofs	441	442	449	1,332
# Bonafide	124	115	108	347
# Videos	565	557	557	1,679

Table 4.2: Statistical information of the WMCA dataset. This information pertains to the complete dataset. The distribution of videos between all sets is equally distributed in thirds ensuring almost equal distribution of different presentation attacks and disjointed set of individuals. Information sourced from [1]

Table 4.2 refers to the statistical information of the full version of the dataset, yet, the version used in this thesis was the trial free version which does not include all of the previous information. The version used in this work has only 850 videos and removes from the dataset the glasses, fake head, rigid mask and paper mask attacks, with 205 bonafide cases and 645 spoofs. Despite the lower number of attacks, this version is still acceptable for the purpose of executing this thesis experiments since it still has more attacks and a greater imbalance between its classes. In a first approach, the distribution between sets was made the same way as the full paid version, yet it was decided that a more conventional distribution of 60% training set, and 20% for both testing and validation sets, was preferable.

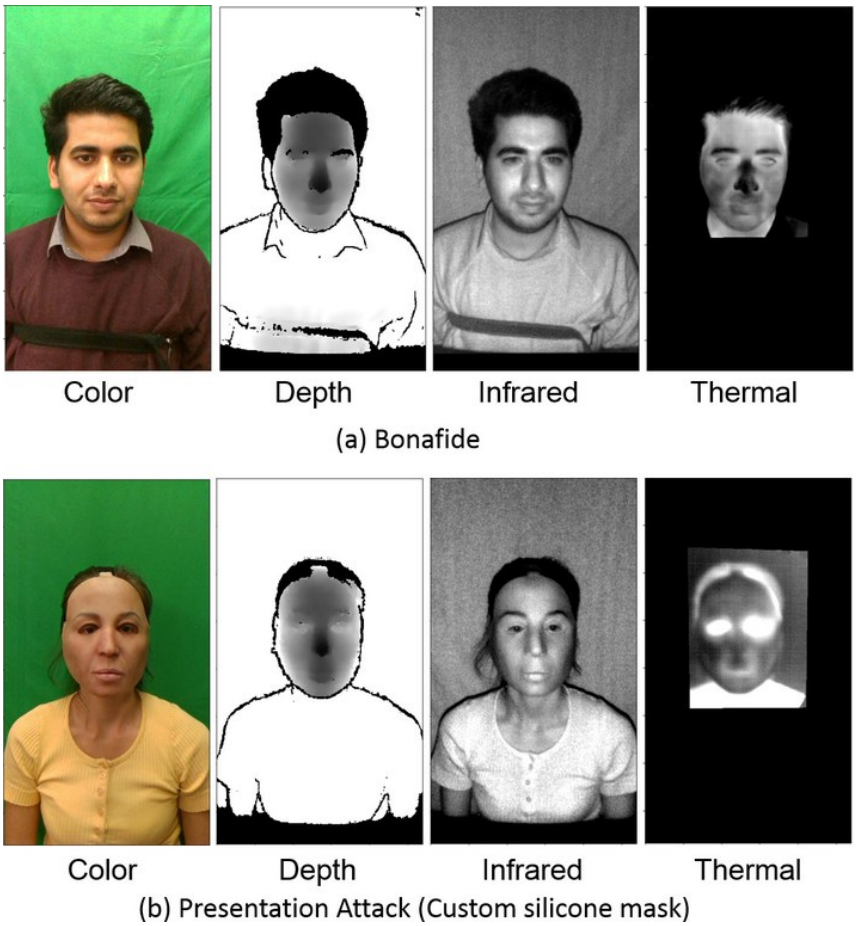


Figure 4.2: Sample images of a) Bonafide and b) Silicone mask attack from the database for all channels after alignment. The images from all channels are aligned with the calibration parameters and normalized to eight bit for better visualization. Taken from [1]

4.2 FeatherNets

FeatherNets was developed by Zhang et al. [22] in the interest of adapting the current deep learning approaches to liveness detection, which are usually very heavy in both computation requirements and data storage, to use in mobile or embedded devices which are incapable of meeting these requirements. To solve this problem, they propose a network ”as light as a feather” that using depth information is able to achieve ACER of 0.00168, with only 0.35 million parameters and 83 million flops down from CASIA-SURF’s baseline using ResNet18 [28] with an ACER of 0.05 with 11.18 million parameters and 1800 million flops.

This network was picked for this thesis because it intends on continuing the effort of reducing the requirements of liveness detection techniques not by working directly in the networks used but the amount of information that is used.

Category	Number of Presentations
Bonafide	205
Print Attack	193
Replay Attack	169
Flexible Mask	283

Table 4.3: Distribution of presentations in the WMCA dataset's free version. The free version removes 4 types of attacks and some examples from the categorys that remain.

	Training	Validation	Testing	Total
# Spoofs	387	129	129	645
# Bonafide	123	41	41	205
# Videos	510	170	170	850
# Frames	25,500	8,500	8,500	42,500

Table 4.4: Statistical information of the WMCA dataset's free version and personal distribution between its training, testing and validation sets. For each video there are 50 frames that unlike CASIA-SURF are not filtered and are all processed.

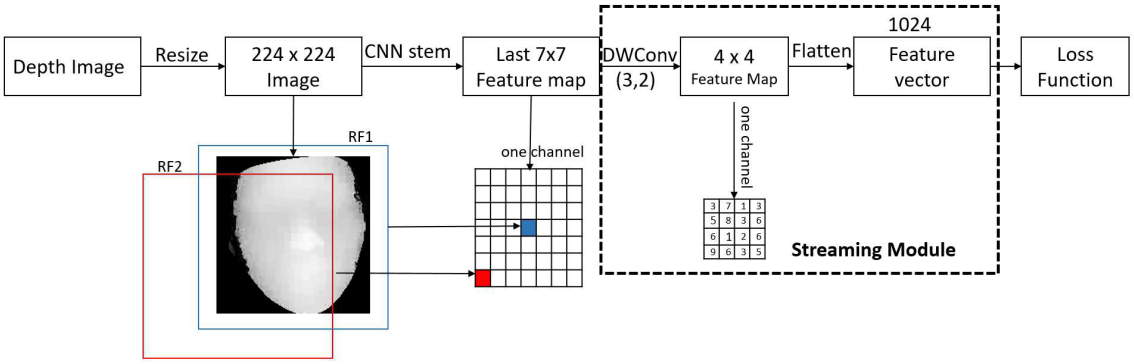


Figure 4.3: FeatherNets' structure. In the last 7×7 feature map, the receptive field and the edge (RF2) portion of the middle part (RF1) is different, because their importance is different. DWConv is used to better identify this different importance. At the same time, the fully connected layer is removed, which makes the network more portable. Taken from [22]

4.2.1 Architecture Design

There are three base building blocks for the construction of the FeatherNets, one main block and two different down-sampling blocks, detailed in the figure below.

4.2.1.1 Block A

The main building block is the inverted residual block proposed by Sandler et al. in "MobileNetV2: Inverted Residuals and linear Bottlenecks" [26] which expands on their previous work "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" [25] that worked with depth wise convolution to greatly reduce the compu-

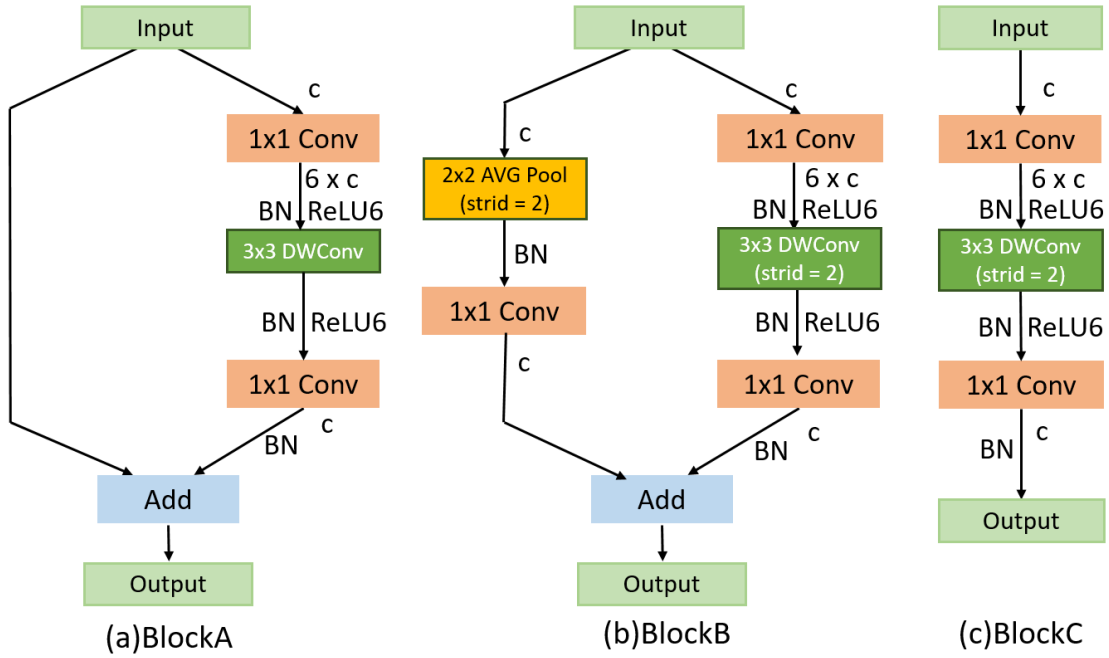


Figure 4.4: FeatherNets' main blocks. FeatherNetA includes BlockA & BlockC. FeatherNetB includes BlockA & BlockB. (BN: BatchNorm; DWConv: depth wise convolution; c:number of input channels.) Taken from [22]

tational cost while only slightly sacrificing the network's accuracy.

Standard convolution filters features based on the convolutional kernels and combines these features to produce a new representation, having the computational cost depending on the number of input channels M, the number of output channels N, the kernel size $D_k \times D_k$ and the feature map size $D_f \times D_f$ resulting in the final cost of:

$$C_1 = D_k \cdot D_k \cdot M \cdot N \cdot D_f \cdot D_f \tag{4.1}$$

Depth wise separable convolution substantially reduces the computation requirements by separating the two phases of filtering and combination using depth wise convolution for the filtering phase which filters each of the M channels separately requiring a point wise convolution to combine them. The final cost of depth wise separable convolution is then:

$$C_2 = D_k \cdot D_k \cdot M \cdot D_f \cdot D_f + M \cdot N \cdot D_f \cdot D_f \tag{4.2}$$

The reduction in computation cost is then equal to:

$$\frac{C_2}{C_1} = \frac{D_k \cdot D_k \cdot M \cdot D_f \cdot D_f + M \cdot N \cdot D_f \cdot D_f}{D_k \cdot D_k \cdot M \cdot N \cdot D_f \cdot D_f} = \frac{1}{N} + \frac{1}{D_k^2} \tag{4.3}$$

The bottleneck blocks present in these networks are similar to residual blocks used

for residual learning [28], with each block having an input followed by several bottlenecks ending with an expansion. However, inspired from the intuition that the bottlenecks contain all the necessary information, and that the expansion is merely an implementation detail, it is possible to use shortcuts between the bottlenecks to allow for better propagation of information with the inverted model being far more memory efficient. The difference between the regular bottleneck and inverted bottleneck blocks are shown in figure 4.5.

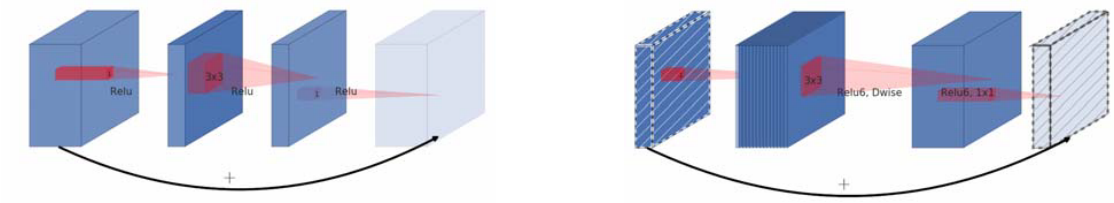


Figure 4.5: The difference between residual block (left figure) and inverted residual (right figure). Note how classical residuals connects the layers with high number of channels, whereas the inverted residuals connect the bottlenecks. Taken from [26]

4.2.1.2 Block B and C

These are the down-sampling blocks for FeatherNet B and FeatherNet A respectively, with block B being the more complex of the two. Block C is almost equal to block A, removing the secondary branch and increasing the stride of the depth wise convolution to 2, thus reducing the dimensions to 12.5% of the input. Block B maintains the secondary branch but adds average pooling (AP) to it which was proven by Szegedy et al. to improve performance by learning more diverse features in Inception [80]. The inclusion of average pooling resulted in an improvement from 0.00261 ACER to 0.00168 ACER, despite this, FeatherNetA was used for the large majority of tests being the simpler approach of the two using less parameters than FeatherNetB.

4.2.2 Streaming Module

The fully-connected layers present in the large majority of CNNs are prone to over fitting, existing better options for the classification portion of the networks. One of these approaches which as since been used in multiple object recognition tasks like the previously mentioned MobileNetV2 is Global Average Pooling (GAP) proposed by Lin et al. [81]. The general idea is to create a feature map for each of the categories present in the classification task and then feeding the average of each feature map directly to the soft-max layer, it has the advantage of the feature maps being more easily interpreted as confidence maps and since there is no parameter to optimize it eliminates the possibility of over fitting at this stage of the network.

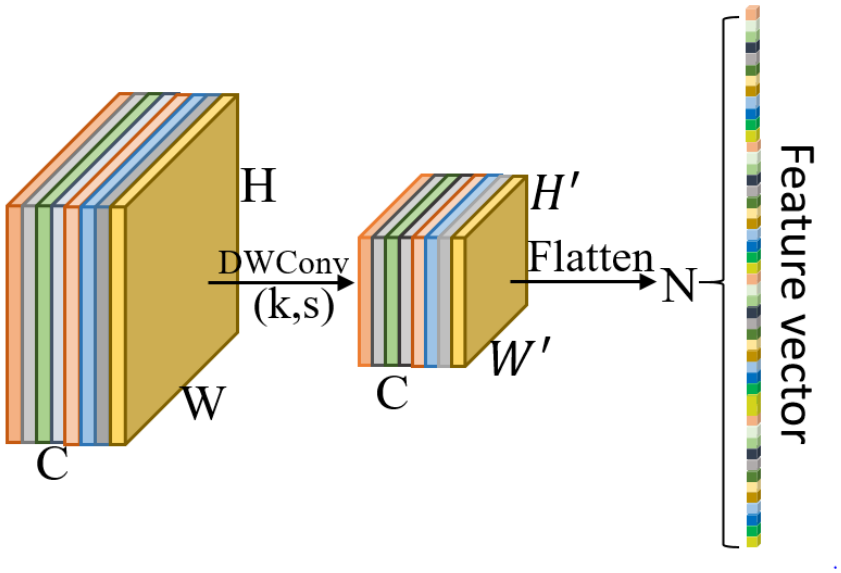


Figure 4.6: Streaming Module. The last blocks' output is down sampled by a depth wise convolution with stride larger than 1 and flattened directly into an one-dimensional vector. Taken from [22]

There have however been cases that show that for facial recognition, networks without GAP have higher accuracy than those with GAP [82, 83], this is due to GAP giving equal importance to all regions of the feature map while for facial recognition tasks, a more Gaussian approach is preferable, note in figure 4.3 the larger effectiveness of RF1's receptive field over RF2's.

To treat the different importances of the feature map, Zhang et al. [22] propose their streaming module, shown in figure 4.6, that results in a feature vector. The vector is calculated by:

$$FV_{n(y,x,m)} = \sum_{i,j} K_{i,j,m} \cdot F_{IN_y(i),IN_x(j),m} \tag{4.4}$$

In the left side of this equation, $FV_{n(y,x,m)}$ stands for the n_{th} element of the $N = H' \times W' \times C$ (height, width and nº of channels of the depth wise convolution's output) elements of the feature vector corresponding to the (y,x) unit of the m_{th} channel and of the output of the depth wise convolution and can be obtained with equation:

$$n(y,x,m) = m \times H' \times W' + y \times H' + x \tag{4.5}$$

In the right side, K is the depth wise convolution's kernel, F is the feature map to which the convolution is applied, m is the channel index, i and j denote the spatial position of the kernel K and $IN_y(i)$ and $IN_x(j)$ denote the corresponding position in the feature map. They can be calculated by:

$$IN_y(i) = y \times S_0 + i \tag{4.6}$$

$$IN_x(j) = x \times S_1 + j \tag{4.7}$$

With S_0 and S_1 being the vertical and horizontal strides respectively. The model was implemented using PyTorch and the final architecture is shown in table 4.5.

Input	Operator	c
$224^2 \times 3$	Conv2d/2	32
$112^2 \times 32$	Block B/C	16
$56^2 \times 16$	Block B/C	32
$28^2 \times 32$	Block A	32
$28^2 \times 32$	Block B/C	48
$14^2 \times 48$	5x Block A	48
$14^2 \times 48$	Block B/C	64
$7^2 \times 64$	2x Block A	64
$7^2 \times 64$	Streaming	1024

Table 4.5: FeatherNets Network Architecture. All spatial convolutions use 3×3 kernels. The factor c stands for the number of channels of the operator's output. Taken from [22]

4.2.3 Focal Loss

The loss function used is the Focal Loss function used in "Focal Loss for Dense Object Detection" [84], developed while attempting to solve the issues present in a scenario of object detection where there is a very large imbalance between the foreground and background classes. It is built upon the basic cross-entropy loss, adding a simple weight balancing parameter to address class imbalance in the dataset, and the focusing parameter in order to down-weight the impact of the decisions made in easy examples; i.e. the more classified categories. The equation is built step by step from binary cross entropy by:

$$CE(p,y) = \begin{cases} -log(p) & \text{if } y = 1 \\ -log(1-p) & \text{otherwise} \end{cases} \tag{4.8}$$

With y specifying the ground-truth class (in the case of this thesis, as presented before, the bonafide class) and $p \in [0,1]$ is the model's estimated probability for the class with label $y = 1$ (the confidence with which the model answers "yes"). For notational convenience, p_t is defined:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \tag{4.9}$$

With this notation, equation 4.8 can be rewritten as:

$$CE(p,y) = CE(p_t) = -log(p_t) \tag{4.10}$$

To address class imbalance, the weighing factor α is added, with $\alpha \in [0,1]$ for class 1 and $1-\alpha$ for class 0. For notational convenience α_t receives an analogous definition to p_t resulting in:

$$CE(p_t) = -\alpha_t log(p_t) \tag{4.11}$$

Finally the focusing parameter is added to focus the training of the model on the "harder" negative predictions. The modulating factor $(1-p_t)^\gamma$ won't then have a great impact on the final loss values for the larger confidence values and affect it more for the lower confidence values. This effect can be regulated with the focusing parameter γ. The final equation for focal loss is then:

$$FL(p_t) = (1-p_t)^\gamma \times CE(p_t) = -\alpha_t(1-p_t)^\gamma log(p_t) \tag{4.12}$$

Adapting the context to the problem at hand, with the positive case being that the recognized individual is bonafide and the false case is that they are spoofed: "Is this face bonafide?", as stated before the model will always answer "yes" with an associated degree of confidence. With focal loss, the more confident the model is that the presented face is bonafide, the less this will affect the model's learning.

4.3 Experimental Settings

This section will detail the various experimental approaches to the problem that this thesis presents. As stated at the beginning of this chapter, none of the approaches employ a modification of the network used itself, instead preferring to work with the parameters used in certain key points and the data used. On the topic of data, all the experiments were made using both the datasets mentioned in the beginning of this chapter, because the differences between them give important insights to the problem at hand.

These differences or more generally, the amount of diversity in a dataset, be it in what is being captured or the conditions the capture is made, are one of the topics that is tested to check on how overfitting can be reduced without even approaching the model used itself. These encompass the majority of the tests made, varying the modality used, the dataset used, and then moving to cross dataset testing. To the parameters themselves, the focusing

parameter of the loss function used and the threshold considered for the labeling decision, the experiments are made to explore how the interpretation of the network's output can influence, directly and indirectly, the final results.

4.3.1 Depth Images Tests

The first conditions are identical to the ones used by Zhang et al. [22], simply to confirm that the results obtained are consistent with the results presented by the authors, and give the initial baseline to which all the following conditions will be compared to. The optimization solver used is Stochastic Gradient Descent (SGD) with a learning rate of 0.001 for both FeatherNet A and FeatherNet B with a decay of 0.1 after every 60 epochs and a momentum setting of 0.9, with FeatherNet A running for 200 epochs and FeatherNet B for 150. The focal loss function is used with $\alpha = 1$ and $\gamma = 3$.

4.3.2 RGB Images Tests

With the intent of eventually applying liveness detection to everyday devices, there can't be a reliance in forms of information not attainable by said devices. As such there is the transition from the depth images used in the initial tests to the RGB images present in the datasets. Again, since both datasets were obtained using the same camera there aren't concerns about differences in quality that could affect the results. Aside from the change in information fed to the model, all other conditions are the same as the ones used initially.

4.3.3 Focus Parameter Tests

For these experiments, the focusing parameter is decreased to 2 and increased to 5 in order to take note on how it affects the results. These values were chosen from the ones used by Lin et al. [84] being the ones closest to the one used by Zhang et al. [22].

4.3.4 Cross Dataset Tests

Cross dataset testing, as the name might suggest, simply entails in testing the model on a different dataset than the one that was use in its training. Being already aware of the differences between CASIA-SURF and WMCA, cross dataset testing was used to check how the model succeeded and how the larger variety of spoofs affects the results, being trained in CASIA-SURF and tested on WMCA and then vice versa.

To further observe the how more spoofs affect a model's performance, a "new" dataset "GRAFTSET" was created by adding, to the initial CASIA-SURF, spoof cases

from WMCA [1]. Only Replay and Mask attacks were added being that Print attacks are already prevalent in CASIA-SURF as it is, and were added by 1%, 5% and 10% of the number of files of CASIA-SURF, initially with only one type of attack added, and then both at the same time. With the new dataset constructed, it was used for cross dataset testing, being used for training with testing being done with WMCA.

4.3.5 Precision-Recall Tests

The final set of experiments relate to the definition of confidence presented in chapter 3 and the tuning of the threshold which separates what should be interpreted as a ”yes” or ”no”. To do so, the method presented by Grandeperrin [85], which consists in the study of a precision-recall (PR) curve, was employed.

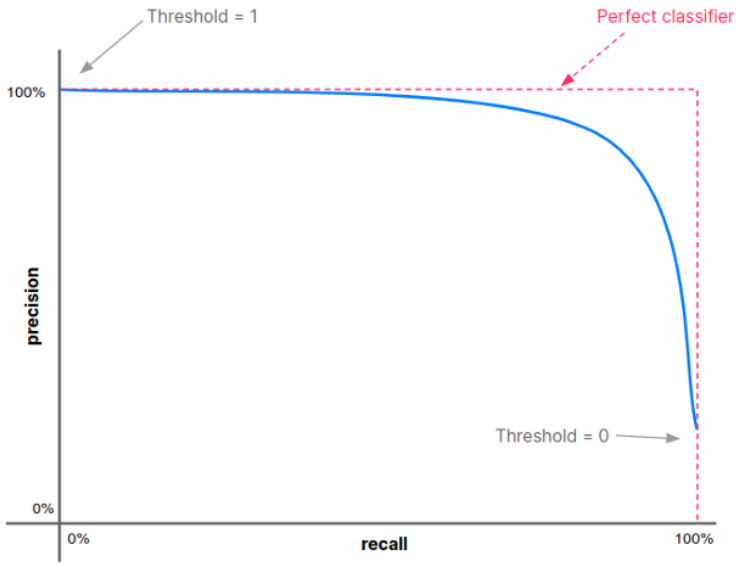


Figure 4.7: Example of a Precision-Recall curve. Taken from [85]

To construct the PR curve, the approach is running the model at different thresholds between 1, where no image can be considered as bonafide and 0 where all predictions will be bonafide. Once all these values are obtained (for this thesis the chosen value is 20) the points can be plotted in a graph and then a curve adjusted to them. From this curve a point can be picked out as what is considered ideal, in this case the closest point to what be considered perfect i.e. $(precision,recall) = (1,1)$, however the threshold value needs to be inferred from where the ideal point stands in the graph. All these tests were conducted using FeatherNet A with $\gamma = 3$ with the threshold values being selected as the experiments went on attempting to achieve the most interesting PR curve.

[1] The name was chosen from the botanical activity of grafting which consists of joining tissues of different plants, for example a branch from an olive tree to the trunk of an apple tree. In this analogy CASIA-SURF is the trunk, and the selected attacks from WMCA are the branches.

Chapter 5

Results and Discussion

From the experimental settings described in the previous chapter, various results were obtained which will be presented in various tables, the first for the depth experiments and the following for the ones using RGB images. Each table will be followed by a brief overlook, followed by the discussion of said results.

Model	Dataset	γ	Best Epoch	Accuracy	ACER
FeatherNet A	CASIA-SURF	2	4	99.063	0.008
		3	4	99.323	0.007
		5	4	98.886	0.012
FeatherNet B	CASIA-SURF	2	4	99.386	0.005
		3	5	99.042	0.010
		5	12	99.178	0.007
FeatherNet A	WMCA	2	16	99.972	0.0005
		3	57	99.958	0.0004
		5	160	99.696	0.005
FeatherNet B	WMCA	2	81	99.986	0.0003
		3	49	99.958	0.0006
		5	122	99.993	0.0001

Table 5.1: Results obtained from depth images. The best epoch corresponds to the epoch that achieved the highest accuracy, not the highest ACER. The value γ is the focusing parameter used in focal loss function.

The results obtained with depth images are all very successful and as such don't leave much room for improvement, they are in line with the results presented by Zhang et al. [22], at least where comparable. The only direct comparison possible is between FeatherNet B with $\gamma = 3$ using the CASIA-SURF dataset to which the result presented was an ACER of 0.00971, most of the other results presented were obtained with their proposed MMFD dataset but have results in the same ballpark. However, from this table it is already possible to draw certain conclusions mostly about the effects of the different

datasets and the effects of the focusing parameter, but these will be discussed in detail once all the relevant results are presented.

Model	Dataset	γ	Best Epoch	EER	Accuracy	APCER	BPCER	ACER
		2	1	0.093	91.996	0.039	0.172	0.105
FeatherNet A	CASIA-SURF	3	9	0.081	89.748	0.129	0.044	0.087
		5	20	0.080	90.466	0.117	0.048	0.082
		2	12	0.093	89.675	0.117	0.073	0.095
FeatherNet B	CASIA-SURF	3	3	0.093	91.674	0.068	0.117	0.092
		5	19	0.067	92.038	0.093	0.049	0.071
		2	16	0.0005	99.972	0.0001	0.001	0.0005
FeatherNet A	WMCA	3	69	0.043	96.988	0.024	0.051	0.038
		5	160	0.004	99.696	0.001	0.008	0.005
		2	81	0.0005	99.986	0.000	0.005	0.0003
FeatherNet B	WMCA	3	63	0.026	98.529	0.009	0.033	0.021
		5	122	0.0003	99.993	0.000	0.0003	0.0001
FeatherNet A	Train CASIA-SURF/Test WMCA	3	1	0.107	90.477	0.050	0.195	0.122
FeatherNet A	Train WMCA/Test CASIA-SURF	3	56	0.032	97.635	0.019	0.039	0.029
	GRAFTSET - 1% Replay		5	0.104	90.416	0.084	0.122	0.103
FeatherNet A	GRAFTSET - 5% Replay	3	11	0.080	89.91	0.126	0.044	0.085
	GRAFTSET - 10% Replay		7	0.089	90.674	0.102	0.072	0.087
	GRAFTSET - 1% Mask		5	0.087	89.489	0.125	0.061	0.093
FeatherNet A	GRAFTSET - 5% Mask	3	18	0.083	88.828	0.144	0.035	0.090
	GRAFTSET - 10% Mask		2	0.106	88.594	0.124	0.091	0.107
	GRAFTSET - 1% Both		0	0.145	87.776	0.069	0.243	0.156
FeatherNet A	GRAFTSET - 5% Both	3	9	0.065	89.913	0.131	0.025	0.078
	GRAFTSET - 10% Both		3	0.087	91.62	0.080	0.094	0.090

Table 5.2: Results obtained from RGB images. Important to note that the ”GRAFTSET” tests are all cross dataset tests with the training with GRAFTSET and testing with WMCA. This table presents EER as an additional metric of success and also presents APCER and BPCER as a means to check if the model fails more in recognising the attacks or the bonafide cases.

Immediately noticeable is the fact that aside from the experiments using only the WMCA dataset, none of the best epochs are ever as high as the ones using depth images by margins of around 10% while maintaining the fact that the best accuracy is obtained in the very early epochs. This could already hint at overfitting but is not a fair assumption since the results of table 5.1 maintain those high accuracy values for the remaining epochs, while this isn’t the case for the RGB images.

With table 5.3 the hypothesis of overfitting is confirmed for all the experiments involving the CASIA-SURF dataset while completely not present in the WMCA experiments. From the very early best epoch (when considering that the models run for 200 and 150 epochs) the suspicion of overfitting is already present, the confirmation comes when looking at the accuracy values presented by the last epochs the model ran, with the

Model	Dataset	γ	Avg. Acc.	Std.	APCER Avg.	BPCER Avg.
		2	52.306	0.888	0.692	0.002
FeatherNet A	CASIA-SURF	3	53.179	0.877	0.679	0.002
		5	60.866	1.422	0.566	0.004
		2	56.582	1.275	0.630	0.002
FeatherNet B	CASIA-SURF	3	58.762	1.170	0.598	0.001
		5	57.667	1.328	0.614	0.002
		2	99.392	0.061	0.005	0.010
FeatherNet A	WMCA	3	95.984	0.165	0.032	0.067
		5	99.449	0.085	0.005	0.008
		2	99.868	0.073	0.001	0.001
FeatherNet B	WMCA	3	97.479	0.298	0.015	0.060
		5	99.917	0.089	0.001	0.0003
FeatherNet A	Train CASIA-SURF/Test WMCA	3	53.272	0.870	0.678	0.001
FeatherNet A	Train WMCA/Test CASIA-SURF	3	96.479	0.250	0.032	0.046
	GRAFTSET - 1% Replay		53.863	0.725	0.666	0.002
FeatherNet A	GRAFTSET - 5% Replay	3	59.541	1.017	0.574	0.004
	GRAFTSET - 10% Replay		59.117	2.061	0.569	0.007
	GRAFTSET - 1% Mask		52.679	0.853	0.684	0.001
FeatherNet A	GRAFTSET - 5% Mask	3	55.249	0.829	0.635	0.003
	GRAFTSET - 10% Mask		59.497	6.063	0.568	0.009
	GRAFTSET - 1% Both		55.048	0.695	0.647	0.001
FeatherNet A	GRAFTSET - 5% Both	3	58.647	0.761	0.577	0.001
	GRAFTSET - 10% Both		59.471	0.492	0.553	0.003

Table 5.3: Average of the 50 last epochs obtained from RGB images. This table presents the averages of the last 50 epochs of each test (epoch 149-199 for FeatherNet A and epoch 99-149 for FeatherNet B) as to display at which values the model settles. The standard deviation of the accuracy average is displayed as to observe the consistency of the results and the APCER and BPCER averages are presented as to be compared to the ones of the best epoch for each experiment to draw conclusions on what is the class with more classification errors.

accuracy values of these epochs being far lower than the one presented for the best epoch. This can be observed in the downwards slope, of some of the graphics presented further in the text. This graphical observation is the interpretation of the ”U-shaped” curve and its translation to an accuracy graph, mentioned in chapter 3. Most of the graphs however, won’t present an inverted ”U” since the values eventually plateau, dragging the shape.

Overfitting is an issue prevalent in many machine learning approaches, and as described in the detailing of FeatherNets architecture, many of their decisions were made precisely to reduce it, but since the model is planned for depth images, these approaches clearly do not translate for the RGB tests. Of course using different methods more adequate for RGB information could be a successful approach, the interest of this thesis is not on adapting or creating a model but to draw conclusions from different details of machine learning. Having all the results obtained from most of the experimental settings described

in the previous chapter available, they can now be discussed.

5.1 Transition from Depth to RGB

Based on the accuracy scores, the use of RGB images is a downgrade from the depth information, more so when looking at the final averages of the model. This isn't an issue of RGB information per se, but the lack of supervision from additional information, as explained in chapter 2 and 3. There are some conclusions to be taken from the fact that, while CASIA-SURF related experiments failed when the transition from depth to RGB occurs and the WMCA experiments maintain their results, but these are more adequate for the comparison between the two datasets, but to explain CASIA-SURF's failure to maintain results is quite simple.

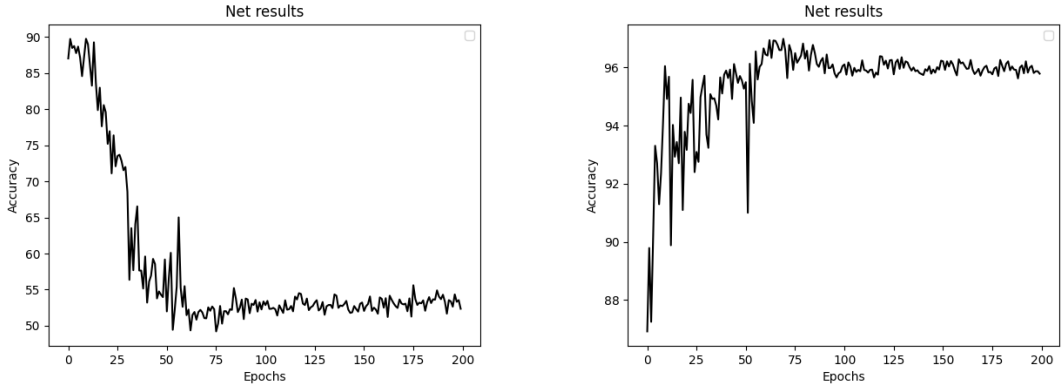
Depth images are capable of giving information that is not very perceptible otherwise, easily spotting attacks that alter the depth of a regular face. Since CASIA-SURF only presents print attacks, which consist in covering an individual's face with a sheet of paper (as far as a depth image is concerned), the model's capability for distinguishing between the two cases is very high. However if the model only has the RGB images, and supposing that the quality of the print is very high, an image of someones face and an image of someone holding someone else's picture might not be as distinguishable, this problem is exacerbated if the image is cropped.

Figure 5.1: Comparison between RGB and depth images of a print attack (left) and a bonafide face (right). Note that the depth images aren't of great quality, not being able to capture the eyes cut out of the print attack and not giving much detail to the bonafide case, but being possible to notice the differences. Images selected from the CASIA-SURF dataset [63].

5.2 Effects of diversity in datasets

For the following discussion it is important to remember the details of both of the datasets used. Not only does WMCA have more attack types, there are also more conditions in which the video recordings take place. Keeping that in mind, and considering

also the issues presented with binary cross entropy loss, that it is prone to overfitting since the model has the possibility of capturing random features not related to the task that was presented. With these considerations, explaining why WMCA shows no overfitting at all while CASIA-SURF's poor final averages indicate that overfitting occurred, consists basically in the fact that even though the problem is still approached with a binary point of view, there is a larger distinction between the bonafide cases, which the model is trying to categorize as such, and the attacks that between them have more differences.



(a) Results from CASIA-SURF **(b)** Results from WMCA

Figure 5.2: Model results from CASIA-SURF and WMCA. Both results were taken in the same conditions using the datasets RGB images. The graphs are almost mirrored with the results from WMCA being what one would expect from a machine learning model.

Despite the differences with the backgrounds between both datasets, it is perhaps unfair to give it much credit since while WMCA does not crop the facial region of the pictures, CASIA-SURF does, as shown in figure 5.1. Such a tactic is employed precisely to reduce the number of unrelated factors possibly present in an image. However the different lighting conditions are a factor that can affect the results presented, but the rest of this discussion will maintain its focus on the amount of different attacks.

To emphasize the effects of more attacks, analyzing the results obtained from the cross-dataset tests which include not only the ones with the basic CASIA-SURF and WMCA but also the ones involving the various GRAFTSETs. The results from the "Train CASIA-SURF/Test WMCA" and "Train WMCA/Test CASIA-SURF" are almost identical to the results obtained from the "CASIA-SURF" and "WMCA" experiments respectably (all experiments conducted with FeatherNet A and $\gamma = 3$), this of course since the training set is maintained and only the testing set is changed. A more diverse training is bound to achieve better results, in fact, Liu et al. [58] developed their dataset SiW-M with 13 different spoof types with the intent of training models to be able to then correctly identify different attack types not present in the initial training set. It is from these conclusions that

the idea for the GRAFTSET tests take place, by adding different spoof cases to the training set of CASIA-SURF there is a slight improvement to the final average of the last epochs of the model. The inclusion of just one type of attack or both achieve similar results in terms of just the average, but having both types of attacks reduces the standard deviation indicating more consistent results.

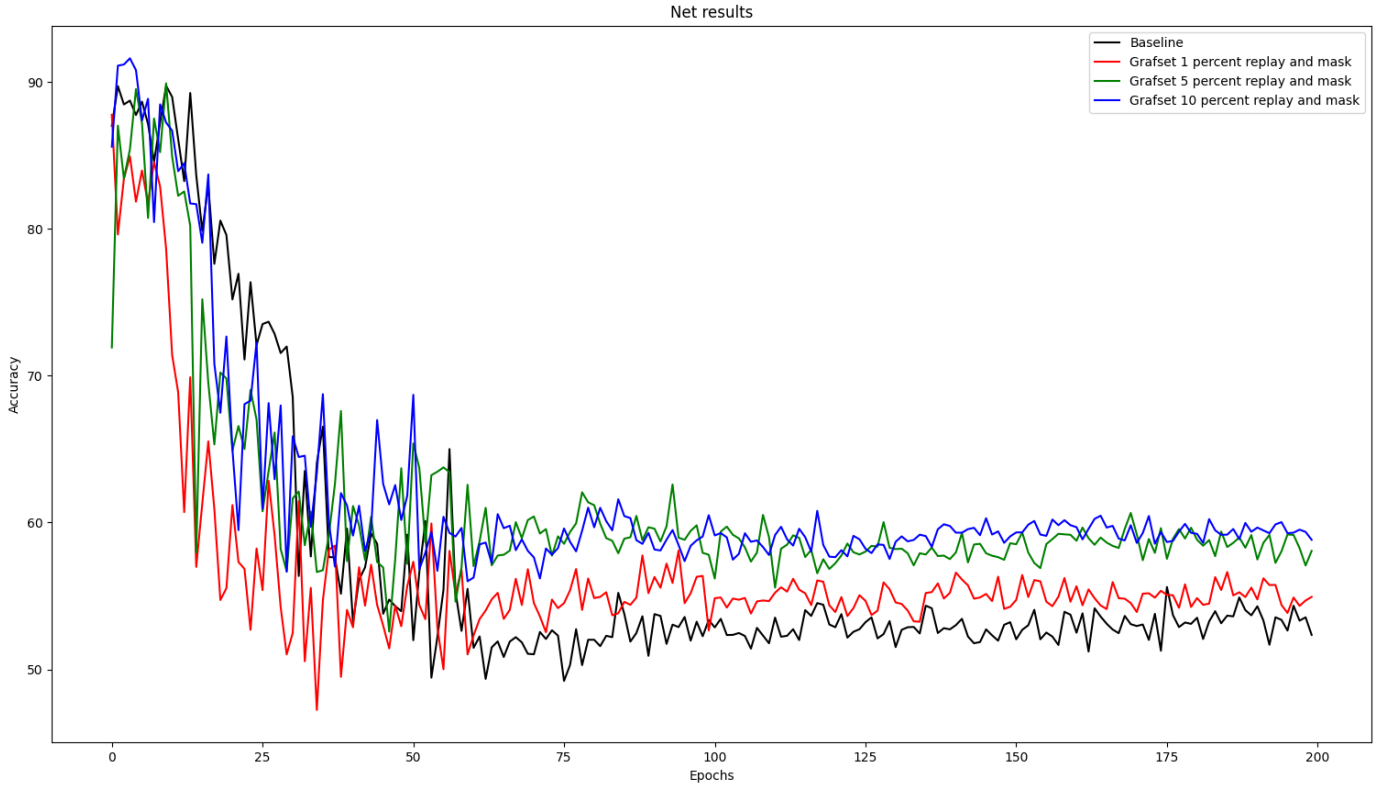


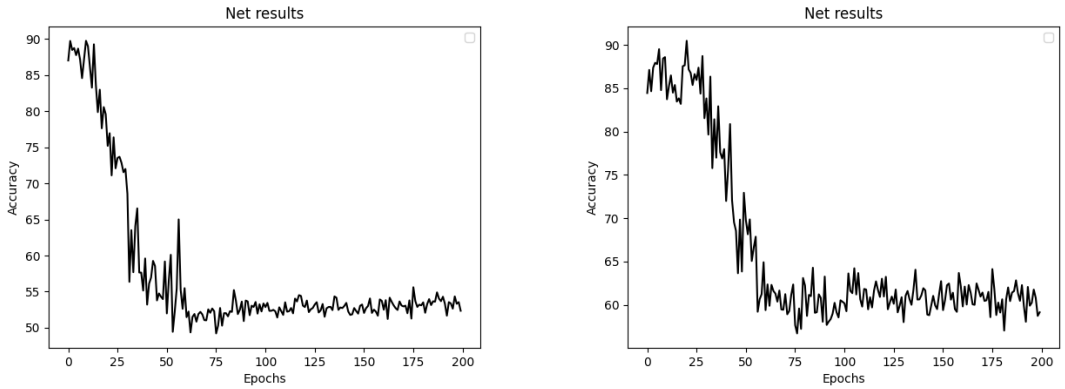
Figure 5.3: Model results for GRAFTSET training with both Mask and Replay attacks. The results obtained for the iterations of GRAFTSET include both types of spoof at 1% (red), 5% (green) and 10% (blue) and are compared to the baseline (black) which are the results obtained from the regular CASIA-SURF tests conducted in the same conditions.

5.3 Effects of the focus parameter

Before discussing the focus parameter itself, the discrepancy between the APCER averages and the BPCER averages has to be addressed. For most experiments, while the BPCER averages are quite low showing very few cases of bonafide cases being labelled as spoofs, the APCER averages are very high reaching values above 50%. This can be considered the worst case scenario since if hypothetically this model would be used for a

security operation, more often than not, an attack would not be detected and a person with perhaps bad intentions would be granted the access that they weren't suppose to have. If the values were inverted with very high BPCER and low APCER, legitimate users would be barred from access to their property but very few successful attacks could occur, a far too restrictive system but secure nonetheless.

Since Focal Loss results in a model that is more focused in the spoof cases and won't learn as much from the what would be considered a bonafide one, it would be expected that it would be able to more successfully categorize spoofs as such. The reality is that through the differences explained in section 5.1, the RGB spoof images don't offer as much as the depth ones and as a consequence, the "focus" is squandered. Reducing the focusing parameter doesn't appear to have much effect on overfitting but increasing it does seem to delay it slightly.



(a) Results from CASIA-SURF $\gamma = 3$ **(b)** Results from CASIA-SURF $\gamma = 5$

Figure 5.4: Model results from CASIA-SURF at $\gamma = 3$ and $\gamma = 5$. The best epoch occurs later when the focusing parameter is higher and the average of the final epochs is higher but considering the also increased standard deviation, it could be argued that these results do not give an adequate conclusion.

To confirm this observation, it's only required to remove the modulating factor from the focal loss equation by turning the focusing parameter γ to 0. Tables 5.4 and 5.5 display these results that when compared to their counterparts using the same datasets and model, are pretty much the same without much improvement or degradation. There is however an observation to be made that without the focusing parameter, the model is still able to achieve great results on the WMCA dataset further solidifying the conclusion that with more different aspects within a dataset, there is less need to implement precautions against overfitting.

Model	Dataset	γ	Best Epoch	EER	Accuracy	APCER	BPCER	ACER
FeatherNet A	CASIA-SURF	0	1	0.094	91.07	0.078	0.115	0.096
FeatherNet A	WMCA	0	108	0.009	99.153	0.009	0.007	0.008

Table 5.4: Results obtained with $\gamma = 0$. With the focusing parameter turned to 0, the model is no longer using focal loss but simply a weighted version of binary cross-entropy represented by equation 4.11.

Model	Dataset	γ	Avg. Acc.	Std.	APCER Avg.	BPCER Avg.
FeatherNet A	CASIA-SURF	0	51.705	0.571	0.701	0.012
FeatherNet A	WMCA	0	98.766	0.148	0.012	0.013

Table 5.5: Average of the 50 last epochs obtained with $\gamma = 0$. These results are presented to comment on how these changes affect overfitting.

5.4 Precision-Recall Curve

To recall the conditions for the Precision-Recall tests, they are made in the same conditions as the first color tests (CASIA-SURF dataset, FeatherNetA, $\gamma = 3$) only changing how confident the model needs to be to consider a face as bonafide through the threshold value, from the results the precision and recall values are calculated and graphed. With the first point already calculated for $threshold = 0.5$ and the two edge points of $threshold = 1$ and $threshold = 0$, the following thresholds were chosen by plotting the simpler PR curves and trying to bridge the gaps between the already present points. Ideally, more points would be calculated in order to achieve a perfectly fitting curve and account for outliers but this was not possible due to how long it would take. These values are detailed in Table 5.6. From these values the final PR curve is then obtained.

5.5 Final Results

With the ”ideal” threshold calculated $threshold = 0.9675$, it is only a matter of repeating the initial experiments of interest with this new value and see if it improves and how.

Immediately noticeable is how the best epoch occurs later over the 200 epochs of FeatherNet A which should already indicate some amount of success in reducing overfitting but is of course not a guaranteed conclusion. Also noticeable is when the model is tested on WMCA there are no false positive predictions demonstrated by $APCER = 0$ while also increasing the false negative cases since the BPCER value increased by quite a lot. Considering such a high threshold value this makes sense, but demonstrates that for different datasets different PR curves should be calculated since the ”ideal” threshold will most certainly vary between them. To confirm if there is no overfitting, once again, the

Threshold	TN	FP	FN	TP	Precision	Recall.	Accuracy
1	6614	0	2994	0	1.000	0.000	68.838
0.99	6587.8	16.2	2644.04	349.96	0.956	0.117	72.312
0.9825	6448.5	165.5	1258.9	1735.1	0.913	0.580	85.175
0.975	5886.12	727.88	493.22	2500.78	0.775	0.835	87.291
0.95	5405.44	1208.56	161.9	2832.1	0.701	0.946	85.736
0.925	4601.4	2012.6	120.64	2873.36	0.588	0.960	77.797
0.9	4401.98	2212.02	73.04	2920.96	0.569	0.976	76.217
0.875	3880.28	2733.72	55.9	2938.1	0.518	0.981	70.966
0.85	3760.78	2853.22	47.66	2946.34	0.508	0.984	69.808
0.825	3233.82	3380.18	27.82	2966.18	0.467	0.991	64.530
0.8	3922.44	2691.56	20.66	2973.34	0.525	0.993	71.771
0.7875	3382.96	3231.04	43.02	2950.98	0.477	0.986	65.924
0.775	3282.98	3331.02	22.06	2971.94	0.472	0.993	65.101
0.75	3088.64	3525.36	21.4	2972.6	0.457	0.993	63.085
0.71	2797.72	3816.28	34.48	2959.52	0.437	0.988	59.921
0.67	2805.36	3808.64	20.62	2973.38	0.438	0.993	60.145
0.5	2120.22	4493.78	4.74	2989.26	0.399	0.998	53.179
0.33	1682.02	4931.98	0	2994	0.378	1.000	48.668
0.25	1482.88	5131.12	1.92	2992.08	0.368	0.999	46.577
0	0	6614	0	2294	0.312	1.000	31.161

Table 5.6: Values used to obtain the Precision-Recall curve. Note that for $threshold = 1$ the precision formula results in a division by 0 and as such would not be valid, the 100% precision comes from the interpretation that since no positive classifications were made, technically none of them are wrong.

average values of the last epochs are presented.

The high averages presented in table 5.8 confirm that in fact, no overfitting has occurred, but the higher standard deviation also indicates that while overall these results can be considered satisfactory, there is a certain degree of variability to the model's results that needs to be considered. The most "stable" and improved results come from the GRAFT-SET experiment which maintains the close results during the later epochs as demonstrated by the lower standard deviation that was only noted when the dataset included both extra spoof types and achieving a lower APCER than BPCER.

Overall, this adaptation resulted in the considerable decrease of the overfitting when it was previously presented, while unfortunately giving worse results for the cases where there was no previous overfitting, keeping in mind that if the threshold tuning was made with WMCA this would not happen but most likely the improvement for the other two would not be so good or would not occur.

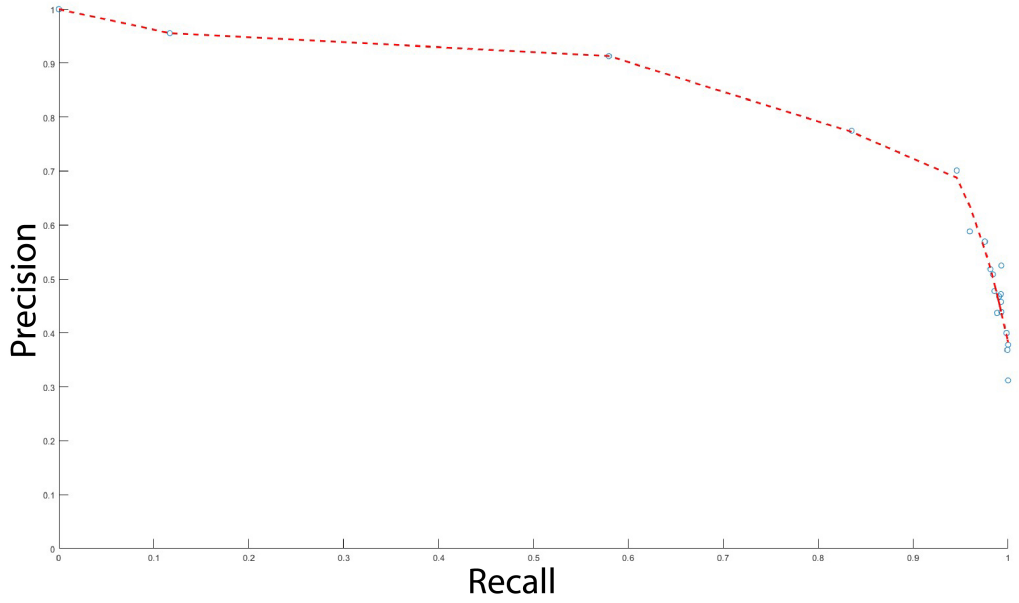


Figure 5.5: Precision-Recall curve. The curve was obtained using Matlab's polyfit() function. The threshold chosen was obtained by using Euclidean distance to find the closest point to the perfect (1,1) which resulted in point (0.8913,0.7828) which corresponds to a threshold value of roughly 0.9675.

Model	Dataset	γ	Best Epoch	EER	Accuracy	APCER	BPCER	ACER
FeatherNet A	CASIA-SURF	3	36	0.117	89.373	0.049	0.232	0.141
FeatherNet A	GRAFTSET - 10% Both	3	59	0.110	90.377	0.046	0.217	0.131
FeatherNet A	WMCA	3	189	0.018	91.165	0	0.385	0.193

Table 5.7: Results obtained with the final threshold. All these experiments were conducted in the same conditions as previously only changing the threshold used.

Model	Dataset	γ	Avg. Acc.	Std.	APCER Avg.	BPCER Avg.
FeatherNet A	CASIA-SURF	3	85.746	1.003	0.160	0.105
FeatherNet A	GRAFTSET - 10% Both	3	87.478	0.401	0.113	0.155
FeatherNet A	WMCA	3	88.446	1.012	0	0.504

Table 5.8: Average of the 50 last epochs obtained with the final threshold. These results are presented to comment on how these changes affect overfitting.

Chapter 6

Conclusion and Future Work

With machine learning being used ever more often for liveness detection solutions, it comes with the problem of overfitting where the model adapts to data incorrectly due to outliers or a minimal set of data. While there are several approaches to attempt to reduce the overfitting effect, these are usually made at an implementation level directly on the model that is constructed. This thesis presented some alternatives more focused in the input and output of the model by approaching the datasets used for the input and the loss function and how the output is interpreted.

These alternatives showed the importance of a varied dataset and how these variations are able to compensate for loss of information associated with the multiple modalities an image can be presented with. From this loss of information, the overfitting effect present in the model became considerably noticeable with a difference between the best result, obtained at epoch 9 with an accuracy of 89.75%, and the average accuracy of the last fifty epoch's, equal to 36.57%. By adjusting the threshold that defined bonafide or spoof, this difference was reduced to 3.63%.

When considering the reason why FeatherNets was chosen for this thesis, the issue can be seen from two different perspectives, a positive one and a negative one. The positive one being that the model was developed with the intent of creating a network "as light as a feather" and as such is able to lower the requirements of liveness detection tasks while maintaining the great results that most modern solutions are able to achieve. On the other hand, there is a problem that comes from the difficult task of generating a perfect model that is optimal in any condition presented to it while maintaining low processing and development cost, a consequence of the concessions made to reduce the processing power and whose consequences are shown when tested in different situations.

The results obtained during this thesis present possible considerations that could be helpful in the development of future solutions, both regarding the size, diversity and appli-

cability of the datasets, as well as the modality given to the model. One of the conclusions that was met is the importance of diverse datasets, which entails that a great benefit to the community would be the development of a dataset that could boost both the quality and dimension of the CASIA-SURF dataset with the number of diverse cases both in presentation attacks and ambient conditions of WMCA. Not only would this dataset be much closer to what a real day-to-day use of a FAS application would encounter, it would also benefit the generalization of models developed with it. Hence meaning, that with a more diverse dataset, the number of studies that deviate from the binary approach to liveness detection by categorizing each attack individually could grow with different insights on what different attacks are more challenging with what modalities.

On a final note, and trying to be straightforward on the best approach regarding the information given to the model, on a regular application, the conclusion was moving away from depth or infra red, on both direct input, or only as a supervision for the model, as well as sticking with the regular color information, proving that the way the model is constructed is of great importance. The building of a new model that, like FeatherNets, tries to be as light as possible, achieving great results and not requiring extra information could benefit from some of the considerations made here. This model would require a new approach to its construction since many of the choices made for FeatherNets were taken considering the depth input. Since this new theoretical model would return to the norm of using RGB images but forego the supervision provided by the extra modalities, techniques that were successful for these types of models might not benefit this one, being perhaps beneficial to consider the approaches used before the extra modalities were available while considering not only the more complex dataset as well as the approaches demonstrated in this thesis.

References

[1] A. George, Z. Mostaani, D. Geissenbuhler, O. Nikisins, A. Anjos, and S. Marcel, "Biometric face presentation attack detection with multi-channel convolutional neural network," *IEEE Transactions on Information Forensics and Security*, vol. 15, 2020.

[2] I. Rigas and O. V. Komogortsev, "Gaze estimation as a framework for iris liveness detection," *IEEE International Joint Conference on Biometrics*, pp. 1–8, Sep. 2014.

[3] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[4] E. Uzun, S. P. H. Chung, I. Essa, and W. Lee, "rtcaptcha: A real-time captcha based liveness detection system," *Proceedings 2018 Network and Distributed System Security Symposium*, 2018.

[5] G. Pan, Z. Wu, and L. Su, "Liveness detection for face recognition," *Recent Advances in Face Recognition*, Jun. 2008.

[6] X. Li, J. Komulainen, G. Zhao, P. C. Yuen, and M. Pietikainen, "Generalized face anti-spoofing by detecting pulse from face videos," *Proceedings - International Conference on Pattern Recognition*, vol. 0, 2016.

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

[8] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[9] Z. Xu, S. Li, and W. Deng, "Learning temporal features using lstm-cnn architecture for face anti-spoofing," *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 141–145, Nov. 2015.

[10] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 864–879, Apr. 2015.

[11] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv preprints arXiv:1408.5601*, Aug. 2014.

[12] H. Hao, M. Pei, and M. Zhao, "Face liveness detection based on client identity using siamese network," *arXiv preprints arXiv:1903.05369*, pp. 172–180, 2019.

[13] X. Xu, Y. Xiong, and W. Xia, "On improving temporal consistency for online face liveness detection," *arXiv preprints arXiv:2006.06756*, Jun. 2020.

[14] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 319–328, Oct. 2017.

[15] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," *CVPR2020*, Mar. 2020.

[16] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 319–328, 2017.

[17] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, 2019.

[18] W. Sun, Y. Song, C. Chen, J. Huang, and A. C. Kot, "Face spoofing detection based on local ternary label supervision in fully convolutional networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3181–3196, 2020.

[19] T. Kim, Y. Kim, I. Kim, and D. Kim, "Basn: Enriching feature representation using bipartite auxiliary supervisions for face anti-spoofing," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 494–503, Oct. 2019.

[20] O. Nikisins, A. George, and S. Marcel, "Domain adaptation in multi-channel autoencoder based features for robust face anti-spoofing," *International Conference on Biometrics, ICB 2019*, Jul. 2019.

[21] H. Kuang, R. Ji, H. Liu, S. Zhang, X. Sun, F. Huang, and B. Zhang, "Multi-modal multi-layer fusion network with average binary center loss for face anti-spoofing,"

Proceedings of the 27th ACM International Conference on Multimedia, pp. 48–56, Oct. 2019.

[22] P. Zhang, F. Zou, Z. Wu, N. Dai, S. Mark, M. Fu, J. Zhao, and K. Li, "Feathernets: Convolutional neural networks as light as feather for face anti-spoofing," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2019-June, 2019.

[23] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 1794–1809, Jul. 2018.

[24] T. Kim and Y. Kim, "Suppressing spoof-irrelevant factors for domain-agnostic face anti-spoofing," *IEEE Access*, vol. September, pp. 86966–86974, 2021.

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, Apr. 2017.

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

[27] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," 2019.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec. 2015.

[29] G. Wang, H. Han, S. Shan, and X. Chen, "Improving cross-database face presentation attack detection via adversarial domain adaptation," *2019 International Conference on Biometrics (ICB)*, pp. 1–8, 2019.

[30] Z. Li, H. Li, K.-Y. Lam, and A. C. Kot, "Unseen face presentation attack detection with hypersphere loss," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2852–2856, 2020.

[31] Y. Qin, C. Zhao, X. Zhu, Z. Wang, Z. Yu, T. Fu, F. Zhou, J. Shi, and Z. Lei, "Learning meta model for zero- and few-shot face anti-spoofing," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 11916–11923, Apr. 2020.

[32] Y. Qin, W. Zhang, J. Shi, Z. Wang, and L. Yan, "One-class adaptation face anti-spoofing with loss function search," *Neurocomputing*, vol. 417, pp. 384–395, Dec. 2020.

[33] R. Shao, X. Lan, and P. C. Yuen, "Regularized fine-grained meta face anti-spoofing," Nov. 2019.

[34] J. L. Wayman, "10 - the scientific development of biometrics over the last 40 years," in *The History of Information Security* (K. D. Leeuw and J. Bergstra, eds.), pp. 263–274, Amsterdam: Elsevier Science B.V., 2007.

[35] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep learning for face anti-spoofing: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, Jun. 2021.

[36] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," *Proceedings - 2012 5th IAPR International Conference on Biometrics, ICB 2012*, 2012.

[37] Z. Yu, J. Wan, Y. Qin, X. Li, S. Z. Li, and G. Zhao, "Nas-fas: Static-dynamic central difference network search for face anti-spoofing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 3005–3023, 2021.

[38] G. Heusch, A. George, D. Geissbuhler, Z. Mostaani, and S. Marcel, "Deep models and shortwave infrared information to detect face presentation attacks," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, pp. 399–409, 2020.

[39] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model.," *ECCV (6)*, vol. 6316, pp. 504–517, 2010.

[40] B. Peixoto, C. Michelassi, and A. Rocha, "Face liveness detection under bad illumination conditions," *ICIP*, 2012.

[41] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," *Proceedings of the International Conference of the Biometrics Special Interest Group, BIOSIG 2012*, 2012.

[42] N. Köse and J.-L. Dugelay, "Shape and texture based countermeasure to protect face recognition systems against mask attacks," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 111–116, Jun. 2013.

[43] N. Erdogmus and S. Marcel, "Spoofing face recognition with 3d masks," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, pp. 1084–1097, 2014.

[44] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, 2015.

[45] A. Pinto, W. R. Schwartz, H. Pedrini, and A. d. R. Rocha, "Using visual rhythms for detecting video-based facial spoof attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1025–1038, 2015.

[46] R. Raghavendra, K. B. Raja, and C. Busch, "Presentation attack detection for face recognition using light field camera," *TIP*, vol. 24, pp. 1060–1075, 2015.

[47] A. Costa-Pazo, S. Bhattacharjee, E. Vazquez-Fernandez, and S. Marcel, "The replay-mobile face presentation-attack database," *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–7, Sep. 2016.

[48] S. Liu, P. C. Yuen, S. Zhang, and G. Zhao, "3d mask face anti-spoofing with remote photoplethysmography," *Computer Vision – ECCV 2016*, pp. 85–100, 2016.

[49] K. Patel, H. Han, and A. K. Jain, "Secure face unlock: Spoof detection on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 2268–2283, Oct. 2016.

[50] J. Galbally and R. Satta, "Three-dimensional and two-and-a-half-dimensional face recognition spoofing using three-dimensional printed models," *IET Biometrics*, vol. 5, pp. 83–91, Jun. 2016.

[51] H. Steiner, A. Kolb, and N. Jung, "Reliable face anti-spoofing using multispectral swir imaging," *2016 International Conference on Biometrics (ICB)*, pp. 1–8, 2016.

[52] I. Chingovska, N. Erdogmus, A. Anjos, and S. Marcel, "Face recognition systems under spoofing attacks," in *Face Recognition Across the Imaging Spectrum*, pp. 165–194, Cham: Springer International Publishing, 2016.

[53] I. Manjani, S. Tariyal, M. Vatsa, R. Singh, and A. Majumdar, "Detecting silicone mask-based presentation attack via deep dictionary learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1713–1723, 2017.

[54] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 612–618, May 2017.

[55] A. Agarwal, D. Yadav, N. Kohli, R. Singh, M. Vatsa, and A. Noore, "Face presentation attack with latex masks in multispectral videos," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 275–283, Jul. 2017.

[56] S. Bhattacharjee and S. Marcel, "What you can't see can help you - extended-range imaging for 3d-mask presentation attack detection," *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–7, Sep. 2017.

[57] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 1794–1809, Jul. 2018.

[58] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[59] M. Liu, H. Fu, Y. Wei, Y. A. U. Rehman, L. man Po, and W. L. Lo, "Light field-based face liveness detection with convolutional neural networks," *Journal of Electronic Imaging*, vol. 28, p. 1, Jan. 2019.

[60] S. Bhattacharjee, A. Mohammadi, and S. Marcel, "Spoofing deep face recognition with custom silicone masks," *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–7, 2018.

[61] S. Jia, X. Li, C. Hu, G. Guo, and Z. Xu, "3d face anti-spoofing with factorized bilinear coding," *arXiv preprint arXiv:2005.06514*, May 2020.

[62] J. Xiao, Y. Tang, J. Guo, Y. Yang, X. Zhu, Z. Lei, and S. Z. Li, "3dma: A multi-modality 3d mask face anti-spoofing database," *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–8, Sep. 2019.

[63] S. Zhang, X. Wang, A. Liu, C. Zhao, J. Wan, S. Escalera, H. Shi, Z. Wang, and S. Z. Li, "A dataset and benchmark for large-scale multi-modal face anti-spoofing," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, 2019.

[64] R. H. Vareto, A. Marcia Saldanha, and W. R. Schwartz, "The swax benchmark: Attacking biometric systems with wax figures," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 986–990, 2020.

[65] Y. Zhang, Z. F. Yin, Y. Li, G. Yin, J. Yan, J. Shao, and Z. Liu, "Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12357 LNCS, 2020.

[66] W. R. Almeida, F. A. Andaló, R. Padilha, G. Bertocco, W. Dias, R. da S. Torres, J. Wainer, and A. Rocha, "Detecting face presentation attacks in mobile devices with a patch-based cnn and a sensor-aware loss function," *PLOS ONE*, vol. 15, Sep. 2020.

[67] A. Liu, Z. Tan, J. Wan, S. Escalera, G. Guo, and S. Z. Li, "Casia-surf cefa: A benchmark for multi-modal cross-ethnicity face anti-spoofing," *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1178–1186, 2021.

[68] A. Liu, C. Zhao, Z. Yu, J. Wan, A. Su, X. Liu, Z. Tan, S. Escalera, J. Xing, Y. Liang, G. Guo, Z. Lei, S. Z. Li, and D. Zhang, "Contrastive context-aware learning for 3d high-fidelity mask face presentation attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2497–2507, 2022.

[69] M. Rostami, L. Spinoulas, M. Hussein, J. Mathai, and W. Abd-Almageed, "Detection and continual learning of novel face presentation attacks," *2021 International Conference on Computer Vision*, Aug. 2021.

[70] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, Dec. 2015.

[71] E. Grossi and M. Buscema, "Introduction to artificial neural networks," *European Journal of Gastroenterology and Hepatology*, vol. 19, pp. 1046–1054, Dec. 2007.

[72] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprints arXiv:1511.08458*, November 2015.

[73] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, Feb. 2019.

[74] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, pp. 15849–15854, Aug. 2019.

[75] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *Proceedings of the Theoretical Foundations of Machine Learning 2017 (TFML 2017)*, Feb. 2017.

[76] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[77] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 319–328, 2017.

[78] A. George and S. Marcel, "Deep pixel-wise binary supervision for face presentation attack detection," *2019 International Conference on Biometrics, ICB 2019*, 2019.

[79] I. Chingovska, A. R. d. Anjos, and S. Marcel, "Biometrics evaluation under spoofing attacks," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2264–2276, 2014.

[80] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Jun. 2015.

[81] M. Lin, Q. Chen, and S. Yan, "Network in network," *Proceedings of the International Conference on Learning Representations 2014 (ICLR2014)*, Dec. 2013.

[82] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholaminejad, J. Gonzalez, and K. Keutzer, "Shift: A zero flop, zero parameter alternative to spatial convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9127–9135, Nov. 2017.

[83] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *arXiv preprint arXiv:1801.07698*, Jan. 2018.

[84] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, 2020.

[85] J. Grandperrin, "How to use confidence scores in machine learning models." `https://towardsdatascience.com/how-to-use-confidence-scores-in-machine-learning-models-abe9773306fa`, 2021. Last accessed 17 August 2022.