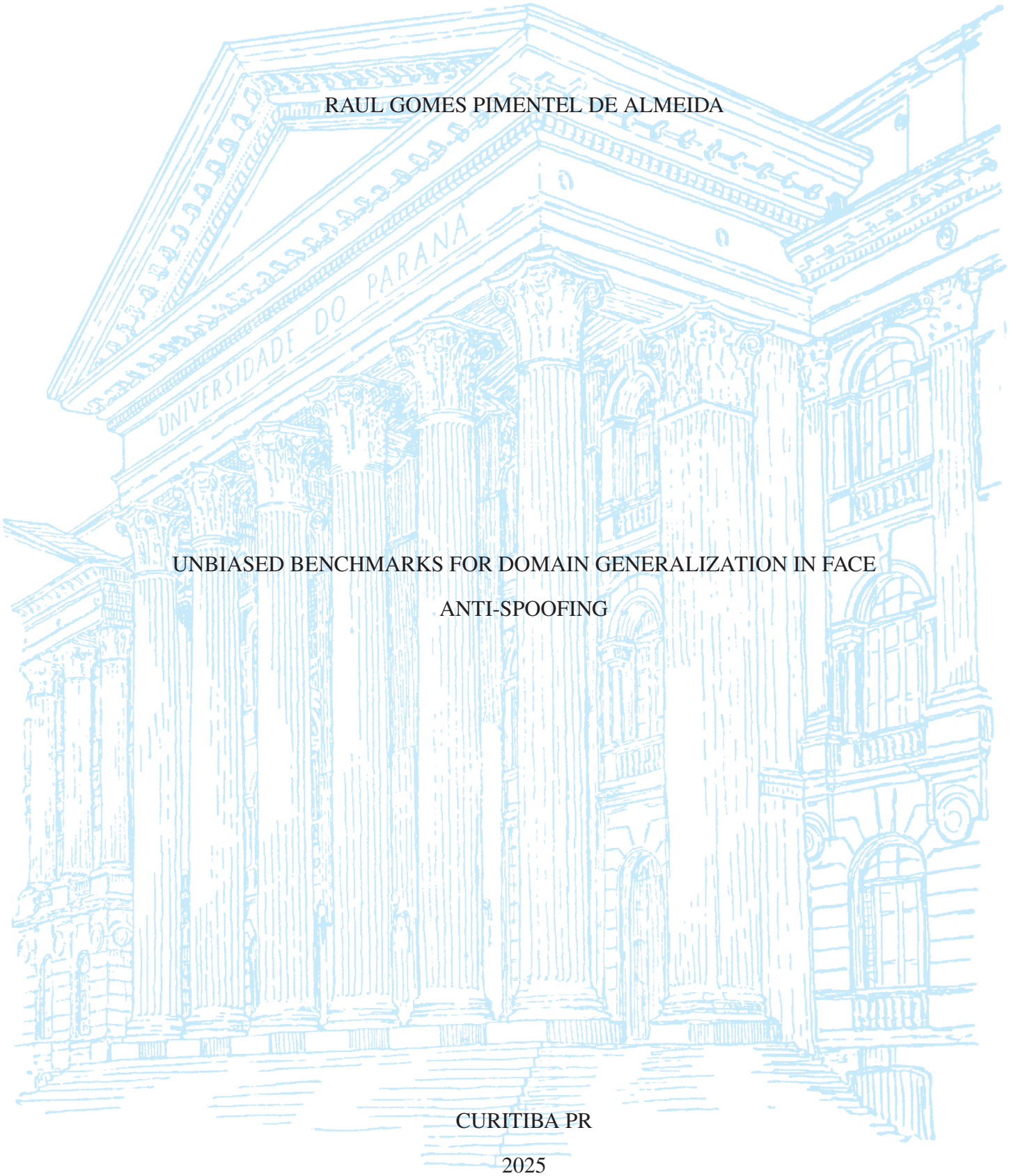
UNIVERSIDADE FEDERAL DO PARANÁ

RAUL GOMES PIMENTEL DE ALMEIDA




# UNBIASED BENCHMARKS FOR DOMAIN GENERALIZATION IN FACE ANTI-SPOOFING

CURITIBA PR

2025

RAUL GOMES PIMENTEL DE ALMEIDA

# UNBIASED BENCHMARKS FOR DOMAIN GENERALIZATION IN FACE ANTI-SPOOFING

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: David Menotti Gomes.

Coorientador: Roger Leitzke Granada.

CURITIBA PR

2025

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Almeida, Raul Gomes Pimentel de
Unbiased benchmarks for domain generalization in face anti-spoofing / Raul Gomes Pimentel de Almeida. – Curitiba, 2025.
1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: David Menotti Gomes

1. Identificação biométrica. 2. Face. 3. Anti-spoofing facial. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Informática. III. Gomes, David Menotti. IV. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

# TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **RAUL GOMES PIMENTEL DE ALMEIDA**, intitulada: **Unbiased Benchmarks for Domain Generalization in Face Anti-Spoofing**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 17 de Março de 2025.

Assinatura Eletrônica
25/03/2025 15:07:35.0
DAVID MENOTTI GOMES
Presidente da Banca Examinadora

Assinatura Eletrônica
25/03/2025 16:00:54.0
RODRIGO MINETTO
Avaliador Externo (UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ)

Assinatura Eletrônica
26/03/2025 10:57:02.0
LUIZ EDUARDO SOARES DE OLIVEIRA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
26/03/2025 11:55:52.0
ROGER LEITZKE GRANADA
Coorientador(a) (UNICO IDTECH)

*A todos os detalhes imprescindíveis à adequada reprodução de experimentos que, cedo ou tarde, no caminho entre a mente dos autores e o texto-fonte dos artigos, foram completamente perdidos*

## AGRADECIMENTOS

"Quando se conserva o amor não se perde a luz"
— Victor Hugo

Qualquer força, constância, dedicação e determinação de que precisei para realizar esse trabalho não partiu de mim, mas de minhas amizades, minha mãe, meu pai, minha irmã - as pessoas que amo e de quem recebo tanto amor.

É a essas pessoas que se deve qualquer inovação, colocação esperta ou perspectiva interessante nas palavras que seguem.

Pelo trabalho, e por todo o amor que vocês me ofereceram ao longo desses anos, eu lhes agradeço profundamente. Amo vocês também.

## RESUMO

*Face spoofing* consiste em simular características biométricas faciais de uma pessoa de maneira a personificá-la em um sistema de reconhecimento facial (por exemplo, em aplicações de pagamento digital e mídia social). Detecção de vivacidade facial ou *face anti-spoofing* (FAS) é o problema de reconhecer ataques como estes. Atualmente o estado da arte nesta tarefa é dominado por modelos de aprendizagem profunda, que requerem uma grande quantidade de dados para seu treinamento. Apesar da dificuldade em se obter estes dados (devido à sua especificidade e sensibilidade), poucos trabalhos se aprofundam no uso de aumentação de dados especificamente para esta tarefa, o que poderia facilitar o enriquecimento do conjunto de treino e fortalecer o desempenho de modelos atuais. Este trabalho propõe uma técnica de aumentação de dados específica para FAS, Landmark Exchange, para melhorar o treino de modelos. Além disso, nós destacamos a presença de viés para o conjunto de teste em *benchmarks* de generalização de domínio (DG) em FAS que ocorre com o uso de dados de teste para validação em cada época. Mostramos que performance em *benchmarks* enviesados não implica em capacidade de generalização, e propomos alternativas não-enviesadas que melhoram resultados do estado da arte.



Palavras-chave: Detecção de vivacidade facial. Face Anti-Spoofing. Detecção de Ataque de Apresentação.

## ABSTRACT

Face spoofing consists of simulating a person's facial biometric traits in order to impersonate them in a face recognition system (for example, in digital payment and social media applications). Face liveness detection or face anti-spoofing (FAS) is the problem of recognizing such attacks. The state of the art in this task is currently dominated by deep learning models, which require large amounts of training data. Despite the difficulty in collecting data (due to its specificity and sensibility), few works explore the usage of data augmentation specifically for this task, even though this could provide easy enrichment of training datasets and strengthen model performance. This work proposes a FAS-specific data augmentation technique, Landmark Exchange, to enhance model training. Furthermore, we shine light on how current Domain Generalization (DG) benchmarks in FAS are biased towards the test set by using test data for validation at every epoch. We show that performance on these biased benchmarks does not imply on generalization capability, and propose unbiased alternatives that enhance results for the state of the art.



Keywords: Face Liveness Detection. Face Anti-Spoofing. Presentation Attack Detection.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ACRÔNIMOS

| | |
|---|---|
| APCER | Attack Presentation Classification Error Rate |
| BPCER | Bonafide Presentation Classification Error Rate |
| FAR | False Acceptance Rate |
| FRR | False Rejection Rate |
| HTER | Half Total Error Rate |
| EER | Equal Error Rate |
| AUC | Area Under the Curve |
| FAS | Face Anti-Spoofing |
| PAD | Presentation Attack Detection |
| GAN | Generative Adversarial Network |
| cGAN | Conditional Generative Adversarial Network |
| ResNet | Residual Network |
| ReLU | Rectified Linear Unit |
| BN | Batch Normalization |
| RGB | Red, Green and Blue |
| SVM | Support Vector Machine |
| IDA | Image Distortion Analysis |
| CNN | Convolutional Neural Network |
| PCA | Principal Component Analysis |
| rPPG | REmote Photoplethysmography |
| RNN | Recurrent Neural Network |
| LSTM | Long Short-Term Memory Network |
| SWIR | Shortwave Infrared |
| SGD | Stochastic Gradient Descent |
| PE | Patch Exchange |
| LMKE | Landmark Exchange |
| PG | Projected Gradient |
| DG | Domain Generalization |
| DA | Domain Adaptation |
| DC-CDN | Dual-Cross Central Difference Convolutional Neural Network |
| SSAN | Shuffled-Style Assembly Network |
| GAC-FAS | Gradient Alignment for Cross-Domain Face Anti-Spoofing |
| SA-FAS | Separability and Alignment Face Anti-Spoofing |
| IADG | Instance-Aware Domain Generalization |

## SUMÁRIO

# 1 INTRODUCTION

In recent years, the use and capacity of mobile phones have grown so as to make these devices an important part of everyday life. This growth was accompanied by a shift in interactions; many activites that could only be done in a computer are now made available in the user's pocket. Important applications followed this tendency, including banking (from Internet banking to mobile applications) and documentation (from government-issued physical documents to verified applications). This was only possible due to accurate verification capability, both in hardware and software, with algorithms such as those of face recognition.

There are, however, strategies for misleading these verification systems, which become easier as the verification becomes decentralized (i.e., the user is not required to be in a physical installation of a company or institution to have their identity verified). For face recognition, many spoofing strategies have been developed and are increasingly honed for improved attacks of impersonation: for example, a malicious invader $A$ could pretend to be some other person $B$ by placing a photo of $B$ in front of their phone's camera in the face verification step of an application's authentication process - the algorithm would recognize the face and let the attacker in. Naturally, access to privileged information and resources should be reserved for privileged users only, and the impact of vulnerabilities to spoofing attacks such as this implies a fragile structure for very essential information exchanges in modern days.

Attacks are the most varied, exploiting weaknesses even in the model's learning process (such as bias and limited representation). In this context, it is fundamental that the task of face anti-spoofing (FAS), also named presentation attack detection (PAD) and liveness detection (here, a face's image is said to be *live* if it is not a spoof), is further researched, so as to enhance current models' performance. Figure 1.1 illustrates two images: one bona fide (real, genuine) and one spoof. In 2024, 24% of the Brazilian population suffered from digital fraud (Agência Senado, 2024).

State-of-the-art FAS models often exploit secondary domain information, such as face depth maps, due to their discriminative advantage over image-domain representations. Face depth maps consist of 3D masks representing a person's face shape. While for bona fide images they maintain this aspect, for spoof images (since there is no face depth) the depth map is empty (a black image). Figure 1.2 exemplifies face depth in the CASIA-FASD (Zhang et al., 2012) dataset for samples of both classes.

It is important to understand that the primary goals of FAS researchers are guided by industry. The only purpose of this task is to ensure that real-world applications are secured against malicious invaders. Because of this, as the state of the art became well-performant in intra-dataset evaluation benchmarks, more focus was directed towards the specific task of Domain Generalization in FAS (DG-FAS), where a model's ability to generalize well to other domains (that is, besides the training dataset) is evaluated. This is important because in real-world applications real authentication attempts (both genuine and fraudulent) are by definition going to be very varied and very different from the training data.

DG-FAS evaluation benchmarks are generally of the following format: the model is trained on the training set and then evaluated on the test set at each epoch. The best test set performance is the one reported as a final result. Due to the difficulty of the task at hand, even with this access to the testing set the state of the art in many benchmarks is far from effective. This bias (in having access to the testing set) is still not ideal, however, and could influence model training and development in negative ways. This will be discussed further and in depth

Figura 1.1: Examples of bona fide (left) and attack (right) images from the CASIA-FASD dataset (Zhang et al., 2012). In particular, the right image is a print attack. Besides more direct traces of spoofs (hands holding a picture and paper distortion), the spoof-presented face has very contrasting texture and color to those found in a bona fide access.

throughout this work, but it is important that the reader reflects on the negative nature of the presence of negative bias in these evaluation benchmarks.

A common method for enhancing model performance both in intra- and cross-dataset tasks is to introduce data augmentation in the training process, so as to expose the trained model to variations in the input images so as to make it more flexible regarding the data that it is capable of correctly classifying. In FAS, to the best of our knowledge, only one work has touched on FAS-specific data augmentation techniques (Yu et al., 2021).

This work follows two paths regarding these issues: one is to explore a new method for data augmentation that is tailored for FAS and another is to explore new, unbiased options for DG-FAS benchmarking while also exposing the issues surrounding the current status quo of model evaluation.

## 1.1 MOTIVATION

Any system that uses face recognition is subject to face spoofing attacks and requires a face anti-spoofing strategy for assurance of its proper functionality and security. Examples of such systems are digital payment and social media applications. At the same time, as prevention methods are improved, attack strategies are also further developed. This adversarial scenario creates the need for continuous development of models that can effectively detect spoofing attempts.

The motivation for tailoring a FAS-specific data augmentation method is to possibly improve the state of the art with a common, albeit underexplored in FAS, technique for enhancing model training (namely data augmentation). Data augmentation is a good option when model training is costly, models do not generalize well or when the training dataset is small - three inherent characteristics of FAS.

At the same time, the motivation for exposing biased benchmarks in DG-FAS and exploring viable unbiased alternatives is to improve evaluation so as to bridge the current gap between academia and industry, which is where these models become in fact useful.

## 1.2 CHALLENGES

In all its importance the FAS task remains a difficult one, particularly due to variety in imaging conditions (variations in light, contrast and color) and attack types among the train and test phases. This same issue with variety presents itself when moving from the train phase to real-world usage, where it becomes aggravated because of production matters such as urgency and reduced ability of data inspection.

The principal aim of research in this field is to understand how we can improve model generalization capabilities with respect to this variety. Some approaches focus on improvements in the models themselves, but model-agnostic techniques could provide pathways towards this aim - one example being data augmentation. This work seeks to improve a previous FAS-specific data augmentation technique (Yu et al., 2021), and one of the largest obstacles to this is reproduction capability, since very little detail is provided by the original authors regarding implementation questions. This applies to the technique itself, but also to model implementation and training, which makes it very difficult to test and compare different methods.

Besides data augmentation, there is the issue of models not adapting well to environments outside of training and testing. Specifically, the bias in current DG-FAS benchmarks allows for misguided model development and might lead researchers and engineers to use models that perform well in all benchmarks but fail to do so in production - which, when the benchmark bias is disregarded, may be attributed only to limitations in training data and in model technology.

## 1.3 PROPOSED APPROACH

A common path for improving a model's performance in FAS is to provide it with additional information about the input image, i.e., to use both the RGB input and its correspondent in another domain where discrimination is easier. One example of such a domain is face depth, where spoofs are modeled to have none and bona fide samples are modeled to have a deep face mask. Figure 1.2 illustrates how genuine and spoof images differentiate in the depth domain.

Previous works report promising results when using face depth information for presentation attack detection (Atoum et al., 2017; Liu et al., 2018a; Shao et al., 2019; Wang et al., 2019, 2020b; Zhang et al., 2020; Zheng et al., 2021; Wang et al., 2021b), as explored in Chapter 3.

Patch Exchange is a data augmentation method proposed in (Yu et al., 2021) that improves the model performance by swapping rectangular regions between samples. In particular, the same rectangular regions are swapped in the face image and in the pseudo depth map, which enforces locality in training the DC-CDN model, a neural network proposed in the context of FAS (Yu et al., 2021). Figure 1.3 illustrates Patch Exchange.

Based on that and on observations of common attacks that have telltale signs in particular regions of the face (see Figure 1.4), we propose Landmark Exchange, where we also swap polygons between samples (just as in Patch Exchange), but these polygons delimit face landmarks, namely eyes, eyebrows, nose, and mouth. Figure 1.5 illustrates the proposed augmentation. We hypothesize that Landmark Exchange can enforce locality just as Patch Exchange does, but focuses on more relevant portions of the image.

In particular, our hypothesis is that Landmark Exchange improves on Patch Exchange in boosting model performance. We verify this hypothesis with experiments on different intra- and cross-dataset benchmarks (see Chapter 5).

Furthermore, we tackle the previously introduced issue of test set bias in current state-of-the-art DG-FAS model benchmarks. Specifically, we propose alternative benchmarks that fundamentally serve the same purpose in measuring model performance, only with the

Figura 1.2: Example face depth maps of bona fide (left) and spoof (right) images from the CASIA-FASD dataset (Zhang et al., 2012). Attacks have an empty depth map by definition, since this is specifically a *face* depth map. However, in the case of attacks where the malicious user's face is partially visible (such as eye masks), the depth map is kept empty. This is the standard in the literature.

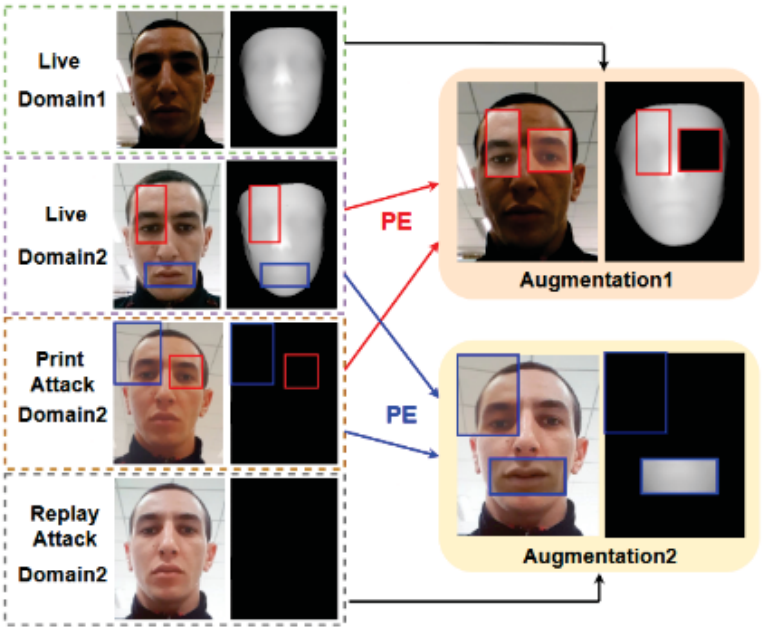



Figura 1.3: Patch Exchange illustration. Pairs of images and depth maps on the left are original samples from the training data, and pairs on the right are samples obtained through Patch Exchange. On the left, pairs *Live Domain1* and *Replay Attack Domain2* are used as basis for the two augmented samples on the right, while the other original samples contribute with patches as indicated by colored arrows. Patches from different samples (pairs of face image and depth map) are exchanged to produce new samples. The exchanges between face images and between depth maps are carried out in the same region for a pair of samples. Note that, even though not a characteristic of these two examples, in Patch Exchange the exchanged regions may consist predominantly of background content. Source: Yu et al. (2021).

test set bias removed. This is done with the introduction of a validation step during training,

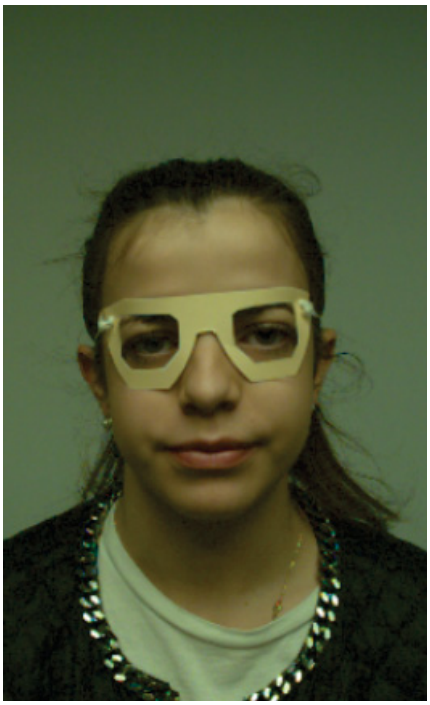


Figura 1.4: An attack that has distinctive characteristics around the eye region. Most of this picture has exclusively bona fide information, with the exception of the paper glasses around the person's eyes. Source: Mostaani et al. (2020).

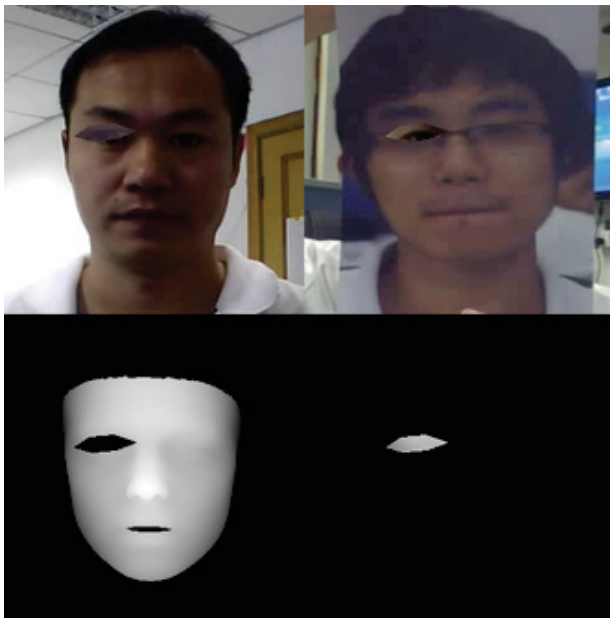
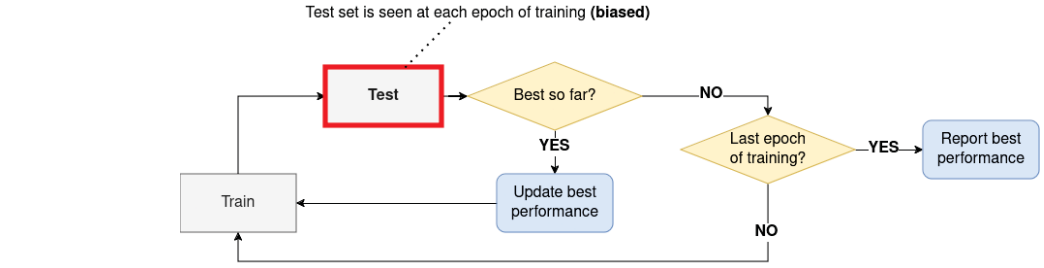


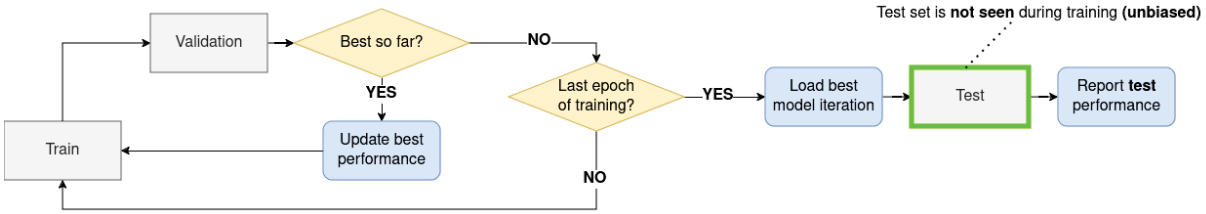
Figura 1.5: Example of two samples generated from Landmark Exchange between a live (left, deep pseudo depth map) and spoof (right, empty pseudo depth map) sample. As in Patch Exchange, the exchange is carried out in both the face images and the depth maps.

where the performance of the model is measured against a dataset that is not seen during training (which means both classification and generalization capabilities are measured) and then the best iteration is picked after training is finished to be tested against the test set. This guarantees that fundamentally the same goals are accomplished - the model's final iteration is the one that has been observed to generalize best - but without any exposure to the test set, thus also guaranteeing

that this generalization capability is not specific towards the test set. Figure 1.6 illustrates both approaches to DG-FAS benchmarking.




(a) Biased benchmark (current standard)




(b) Proposed unbiased benchmark

Figura 1.6: Biased benchmark (top) vs unbiased (bottom). In the biased benchmark, there is access to the test set in every epoch of training, which leads to biased decision-making (both in the research process and model choice for final evaluation). In our proposed unbiased benchmark, the test set is only used after training is finished, so there is no bias in decision-making. Note that this is the same as Figure 4.17.

Our hypothesis regarding bias in DG-FAS benchmarks is twofold: first, that good performance in the current biased benchmarks does not indicate generalization capability; and second, that our proposed unbiased alternatives have the opposite effect.

To the best of our knowledge, this is the first work to propose face landmark-based data augmentation for FAS, and also the first work to propose unbiased DG-FAS benchmarks.

## 1.4 OBJECTIVES

This work's objectives are twofold, guided by both Landmark Exchange and the proposed unbiased DG-FAS benchmarks in answering the following questions - which are further developed in the next chapters, with their conclusions discussed in Chapters 5 and 6:

- How does Landmark Exchange affect a classifier's performance, both in intra- and cross-dataset scenarios?
- How do models that do not rely on depth learn with the proposed augmentation?
- How do state-of-the-art DG-FAS models trained in state-of-the-art (biased) DG-FAS benchmarks perform when subject to new domain changes?
- How do state-of-the-art DG-FAS models trained in our proposed unbiased DG-FAS benchmarks perform when subject to new domain changes?

## 1.5 HYPOTHESES

To summarize, our hypotheses are as described below.

- That Landmark Exchange shows better results in improving model performance in comparison to Patch Exchange;
- That performance in current biased DG-FAS benchmarks is not indicative of model generalization capability;
- That, contrary to widespread biased DG-FAS benchmarks, performance in our proposed unbiased alternatives **is** indicative of model generalization capability.

These hypotheses are verified in later chapters.

## 1.6 CONTRIBUTIONS

We successfully explore each of the questions and experiment with the proposed approaches in different benchmarks and scenarios with four different recent models. In particular, we bring forth the following specific contributions:

- Introducing a novel approach to FAS-specific data augmentation;
- Providing an unbiased alternative that is coherent with the community's directions in terms of dataset preference and evaluation format;
- Improving the performance of the state-of-the-art method GAC-FAS (Le e Woo, 2024) in DG-FAS by removing (and not adding) unfair advantages during training.

Furthermore, we provide public code access[1], with the aim of allowing complete reproducibility of reported results, and benchmark DG-FAS state-of-the-art models against WFAS (Wang et al., 2023), a huge in-the-wild FAS dataset.

## 1.7 OUTLINE

The chapters in this work are organized as follows. Chapter 2 describes fundamental concepts and metrics related to liveness detection. Chapter 3 presents a study of related works. Chapter 4 presents this work's proposed approaches, which is evaluated as described and presented in Chapter 5. Limitations and possibilites for future work are finally discussed in Chapter 6.

[1] `https://github.com/BOVIFOCR/unbiased-benchmarks-aggregator`

# 2 THEORETICAL BACKGROUND

Relevant concepts and metrics related to the field of face liveness detection and networks used in Chapters 4 and 5 are now presented. A foundational understanding of machine learning is expected from the reader. This section's structure is heavily inspired by a 2022 survey (Yu et al., 2022) on Deep Learning for Face Anti-Spoofing. We start by presenting the concept of Face Spoofing Attacks along with a general description of benchmark design in FAS. We then move on to presenting a taxonomy of DL-based methods for FAS in the context of acquisition with commercial RGB cameras, and briefly discuss the case of advanced sensors.

## 2.1 SPOOFING ATTACKS

Face Spoofing attacks, usually called Presentation Attacks (PAs) or spoofs, are attempts to fool face recognition systems by introducing unexpected artifacts in the input captured by the system's sensor (e.g., a commercial RGB camera). This can be as simple as placing a printed photo of someone's face in front of their mobile phone to unlock the device with face recognition, but there are very sophisticated techniques for intrusion. Besides impersonating someone else, there is also the category of obfuscation attacks, where the goal is only to hide the malicious user's own identity. This can be achieved with makeup, for example.

Another possible categorization for PAs is between 2D and 3D attacks. 2D attacks involve a plain attack instrument, such as paper or a screen. This includes replay attacks, when the malicious user places a screen with a playback of a video of someone else in front of the sensor. 3D attacks, on the other hand, involve the instrument's depth in the attack process. An example of this is a silicon mask. 2D attacks are more practical and affordable, but 3D attacks can be more efficient due to their greater similarity in attributes to a human face, such as depth and texture.

Finally, one can divide PAs between whole or partial attacks, with respect to how much of the attacker's face is covered.

There is another category of attacks that is not in focus in this work: digital manipulation attacks. These fool the system with visually imperceptible changes in a virtual domain. As these attacks are performed virtually, they do not fit as presentation attacks: a direct usage of a digitally manipulated picture could be to present it in front of a camera from paper, but that would be a print attack - when the malicious user places a printed picture of someone in front of the sensor. For this reason, besides attacking a system's face recognition method, this technique also relies on infiltrating a system's sensor capture method (to use something other than the device's camera as input, for example).

Face Anti-Spoofing may happen either in parallel or sequentially before the Face Recognition step, i.e., the input is checked for attacks either before or while a person is identified.

Other sensors besides RGB cameras may be used to enhance the classification capability. Due to the widespread nature of RGB sensor usage, however, this is the area that receives the most attention in current research, and that is the case in this work as well. It must be noted, however, that this widespread usage is not due to some technical superiority of RGB sensors over other technologies, but instead because they are vastly present and of practical use. Specialized sensors can make it a lot easier to bridge domain gaps between train and test data, i.e., differences such as recording environment, subject behavior, etc.

## 2.2 LOSS FUNCTIONS

Two loss functions are very common in methods mentioned in this work and in this work itself. They are Binary Cross-Entropy and Mean-Squared Error. We now briefly present each of them.

### 2.2.1 Binary Cross-Entropy (BCE)

When tackling FAS as a binary task, it is common to assign the positive class to bona fide samples and the negative class to spoofs, even though the contrary is sometimes done as well. We use Cross-Entropy loss to accumulate in a single metric a sum of errors of each pair $(x, y)$ of predicted probability of belonging to the positive class $x$ and ground truth label $y$. The Binary Cross-Entropy loss function is a particular (binary) case of the Cross-Entropy function which can be very simply and conveniently represented as

$$BCE(x, y) = -y \times \log x + (1 - y) \times \log (1 - x), \tag{2.1}$$

where $x \in \mathbb{R}_+,\ x \leq 1$ and $y \in \{0, 1\}$.

### 2.2.2 Mean-Squared Error (MSE)

Besides the direct binary task modelling of FAS, many methods utilize auxiliary tasks that involve outputting some image that has a direct correspondence to the input image, such as depth maps. This will be explained later in this chapter. The Mean-Squared Error loss function can be seen as a pixel-by-pixel error mean. For a series of predictions $X$ and a series of corresponding labels $Y$ as similarly defined in Subsection 2.2.1, the MSE loss is calculated as follows:

$$\text{MSE}(X, Y) = \sum_{i=0}^{|X|} (X_i - Y_i)^2, \tag{2.2}$$

where $X_i, Y_i \in \mathbb{R}$.

## 2.3 EVALUATION METRICS

Systems usually focus on the bona fide vs spoof binary task, with other labels being used only in auxiliary tasks for enhanced training. For this reason, and since FAS suffers from great class imbalance (i.e., there are a lot more bona fide samples than spoof samples), the False Acceptance Rate (FAR), False Rejection Rate (FRR), Half-Total Error Rate (HTER), Equal Error Rate (EER) and Area Under Curve (AUC) metrics described below are used in system evaluation.

FAR and FRR (Galbally et al., 2012) correspond to the rate of spoofing attacks incorrectly recognized as live accesses and the rate of live accesses incorrectly recognized as spoofing attacks, respectively. They are computed from the count of false positives (FP), false negatives (FN), true negatives (TN) and true positives (TP) as in Equations 2.3 and 2.4. HTER (Chingovska et al., 2014) is the mean between FAR and FRR, i.e., HTER $= \frac{\text{FAR+FRR}}{2}$.

$$\text{FAR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{2.3}$$

$$\text{FRR} = \frac{\text{False Negatives}}{\text{False Negatives} + \text{True Positives}} \tag{2.4}$$

EER (Ramachandra e Busch, 2017) is a specification of HTER, in that it measures the mean between FAR and FRR at a threshold where FAR = FRR. In practice, with the specifics of the data and model being used, there may be no such threshold, and so the threshold that minimizes |FAR − FRR| is chosen.

AUC (Hanley e McNeil, 1982) is the area under the Receiver Operating Characteristic (ROC) curve. It is related to metrics which test whether positives (in our case, bona fide samples) are ranked with a greater probability of being live accesses than negatives (spoofs) (Hanley e McNeil, 1982).

Even though the metrics above allow for some understanding of how a method performs in the FAS task, when considering a range of different attack instruments the model might perform very differently for each one. For this reason, the APCER, BPCER and ACER (30107-3:2023, 2023) metrics are also used. They are calculated similarly to FAR, FRR and HTER, respectively, but for each attack instrument, and the final chosen value is the worst case, i.e., the largest value of ACER among all presentation attack instruments.

## 2.4 EVALUATION BENCHMARKS

There is a very simple way to divide benchmarks in 4 categories regarding the relation between datasets and attack types in the train and test sets. We describe them below.

**Intra-dataset intra-type:** benchmarks for evaluating a model's discrimination ability under scenarios with very slight domain shifts, since the test and train data are sampled from the same dataset. This means they share similar domain distributions. This is the simplest of the four categories and in many cases DL models have a very satisfying performance in this type of scenario.

**Cross-dataset intra-type:** benchmarks for evaluating a model's ability to generalize to other domains, i.e., to perform well against the same type of attack but large domain shifts.

**Intra-dataset cross-type:** benchmarks for evaluating a model's ability to generalize to unknown attack types with only slight domain shifts, i.e., when sampling train and test from the same dataset. This is usually executed in a leave-one-attack-type-out manner.

**Cross-dataset cross-type:** this is the most challenging of the four categories, and involves both different domains and different attacks between train and test data. An example of this is using datasets with 2D mask attacks for training and datasets with 3D masks for testing.

There are other trends about practical benchmark settings that may make some of them more specific than others, but this general categorization strategy is widely used and draws a clear line between benchmarks.

It is of particular importance to this work that the reader understands how the cross-dataset (i.e., DG-FAS) benchmarks are carried out. They are very similar to what is common in many other deep learning-dominated fields, where a model is trained for a number of epochs with a training dataset and a validation dataset (and the latter is used for model evaluation only, and not for weight updates) and then afterwards tested on a test dataset. Most importantly, these three datasets are disjoint - training, evaluation and testing data are all different. In particular, the model is tested against the validation set every few epochs (or even every epoch) to aid researchers

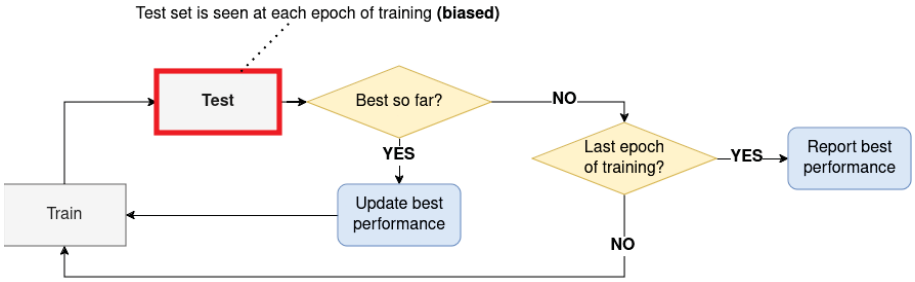



Figura 2.1: Model training in current DG-FAS benchmarks (biased).

in training-related decision-making, and the test dataset is only accessed after training is finished and the state of the model is frozen.

As the goal in DG-FAS is, after all, to achieve domain generalization, the training and testing sets used are from different domains - usually different datasets, from different sources. What is particular of DG-FAS, however, is that there is no validation set. Instead, the testing set is used as validation during training: after each epoch, the model is evaluated against the test set. After training is completed, the best performance across all epochs is the one that is reported as a result. Figure 2.1 illustrates this process. Note that this exposure to the test set during training and the cherrypicking of the reported performance are both biased approaches to model evaluation and hurt the quality of evaluation itself. As a consequence, the model development work is more disturbed by bias-induced noise, and models considered state-of-the-art may be less reliable for real-world usage.

## 2.5 TAXONOMY OF DL-BASED FAS METHODS

Inspired by Yu et al. (2022) in the classification of DL-based FAS methods, we briefly go over methods as categorized in Figure 2.2.

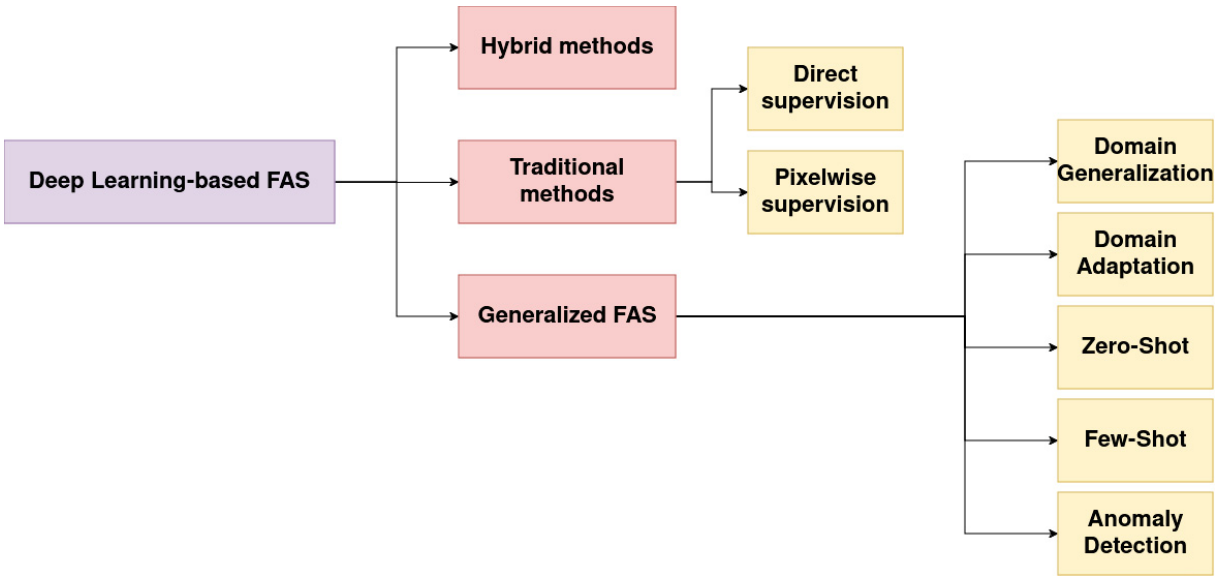



Figura 2.2: Topology of deep learning FAS methods.

**Hybrid methods:** these methods combine handcrafted and deep features. Although DL has shown great promise in many CV tasks, it is easy for methods to overfit in FAS due to the limited nature (both in quantity and in diversity) of data. Meanwhile, handcrafted features

(such as LBP (de Freitas Pereira et al., 2013) and HOG (Komulainen et al., 2013)) have been proven to be discriminative in distinguishing bona fide from PAs in many scenarios. Hybrid works combine these two types of features for input processing and classification.

**Traditional DL methods:** these methods use end-to-end CNNs to learn a mapping from face images to the binary bona fide/spoof label. They have become effective from the development of advanced CNNs, regularization techniques and large-scale FAS datasets and now dominate the field. These usually include direct supervision with BCE, pixel-wise supervision with auxiliary tasks or generative models.

*Direct supervision* is a very straightforward way to apply CNNs to this task, but the losses typically involved only provide global constraints for live/spoof embedding learning and can lead to overfitting on arbitrary clues. Another issue with this approach is that models trained in this framework can be very difficult to be interpreted and end up being used as black boxes.

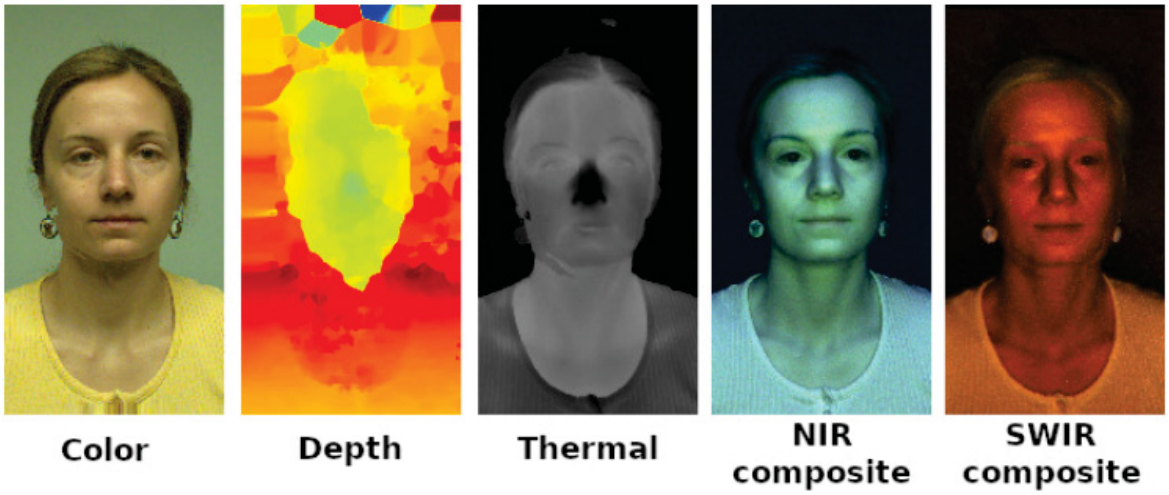



Figura 2.3: Examples of different sensors for capturing face images. These variations can often better highlight different liveness clues. Note that the depth example does not consist of pseudo depth as discussed in the rest of this work, but of an actual depth depiction of the whole figure. Source: Mostaani et al. (2020).

*Pixel-wise supervision,* on the other hand, can provide more fine-grained and contextual task-related clues. There are two approaches for pixel-wise supervision: (1) an auxiliary task describing local live/spoof clues with pseudo depth, binary mask, reflection maps, and other (see Figure 2.3) or (2) estimating a generic spoof patterns with generative models, in an explicit pixel-wise supervision strategy (e.g., reconstructing the original face).

The first option benefits explainable features, since one understands very easily what the model got wrong or right about the tasks from examples, but require high-resolution training data, suffer effectiveness drops when training data is too noisy and have either human-designed or off-the-shelf-algorithm-generated labels, neither of which is a completely reliable method.

The second option, on the other hand, relaxes the need for expert intervention in creating labels, and intuitively allows the model to learn more image-intrinsic spoof-related traces. The predicted spoof patterns are strongly data-driven and can be very insightful visually, but challenging to describe with human rationale. Since this approach also relies on end-to-end learning, it also suffers the risk of overfitting on irrelevant noise (i.e., poor generalization).

**Generalized DL methods:** the method categories previously described all suffer from a very fundamental pitfall - they can be very sensitive to changes in domain. This makes it rather inadequate to apply them in real life. Generalized DL research tries to bridge that very gap, and methods may be divided in four categories described next.

*Domain adaptation and generalization:* these two categories involve methods that aim to generalize to unseen domains, the first using unlabeled test data to improve the model's classification ability and the second using only different training strategies to create a generalizable model. In the context of domain adaptation, it is difficult to collect a lot of unlabeled target data (especially spoofs) for training, besides there being a privacy issue in using the source (train) data during the adaptation step in deployment. Domain generalization models are more attractive since they do not require test data, but it is still unknown how generalization affects the discrimination capability of a model (for spoof detection) in the seen scenarios.

Treating FAS as a closed-set task is very common, but the reality is that the nature of PAs is ever-evolving and real-world applications cannot rely on models only prepared to detect simple, well-known attacks. The other two categories focus on generalizing to unknown attack types. *Zero-* and *few-shot* methods learn to detect novel attack types with either no or very few samples. *Anomaly detection* is a very intuitive strategy for generalizing well to other attack types because the goal there is to learn a compact representation for bona fide samples and then treat spoofs as anomalies. It is intuitive because, while we cannot anticipate all types of attacks that will be used in the future, we already know and have access to bona fide characteristics. Anomaly detection methods have a satisfactory generalization capacity but still suffer from discrimination capability degradation.

# 3 RELATED WORK

This chapter explains all the datasets and methods proposed in studied works. Current challenges in the field are discussed, as are the difficulties with datasets.

## 3.1 DATASETS

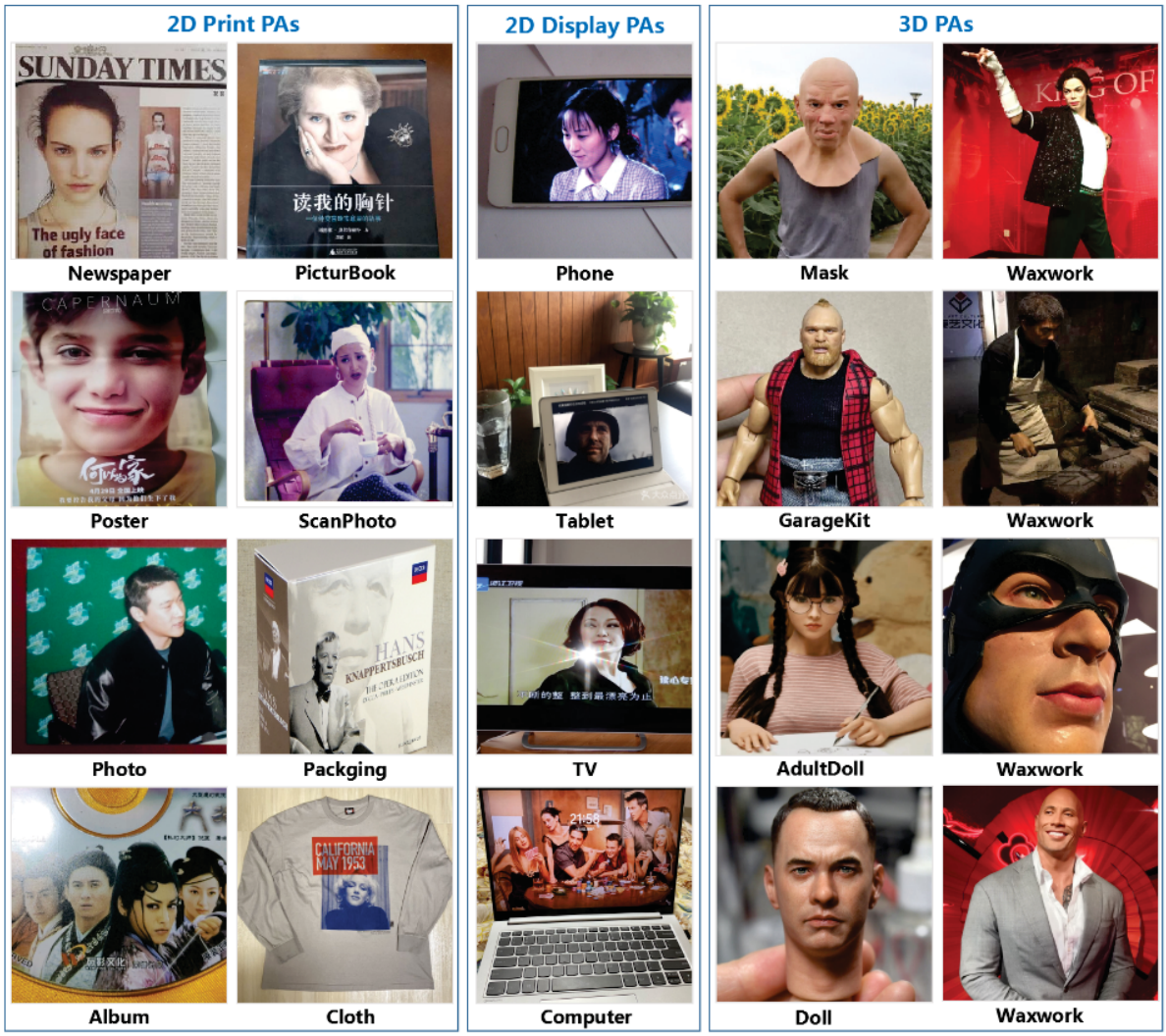



Figura 3.1: Example spoofs from the WFAS dataset. These spoofs are not created from the live samples, but instead they are scraped examples from the internet. Source: Wang et al. (2023).

With the growth in the field of face liveness detection research, there have been many different approaches to dataset collection and what matters in a dataset. For some works, for example, multiple image modes are important, as many methods take advantage of information external to what an RGB image can entail. Currently, there is a focus on sample abundance and variability in acquisition conditions (such as subject movement, camera quality and environmental light), subject characteristics (gender, race, etc) and attack methods. Domain Generalization benchmarks use multiple datasets for training and testing and so are a way to bridge this gap.

Recently, the Face Anti-Spoofing in the Wild (WFAS) (Wang et al., 2023) dataset was published, with great abundance and variability in comparison to previous datasets.

(a) Live WFAS sample where the face is completely visible and very close to directly turned towards the camera.

(b) In some WFAS live samples, the face is partially hidden by body parts or objects, as in this bona fide sample.

Figura 3.2: Examples of live examples from the WFAS dataset.

It is important to notice, however, that the spoofs in WFAS are mostly non-intentional (images scraped from the internet that were not *meant* to be spoofs, but *could* be used as such), and so in some specific aspects it could be less challenging than other datasets. In general, again, WFAS makes up for it with abundance and variability. Figure 3.1 shows spoof examples from the WFAS dataset, and Figures 3.2(a) and 3.2(b) show live examples. Table 3.1 summarizes the studied datasets.

| Name | Year | Citation | #Real | #Attack | Subjects | Attack types | Additional description |
|---|---|---|---|---|---|---|---|
| NUAA | 2010 | Tan et al. (2010) | 5105 | 7509 | 15 | 1 | |
| PRINT-ATTACK | 2011 | Anjos e Marcel (2011) | 200 | 200 | 50 | 1 | |
| CASIA | 2012 | Zhang et al. (2012) | 150 | 450 | 50 | 3 | |
| Replay-Attack | 2012 | Chingovska et al. (2012) | 200 | 1000 | 50 | 3 | Different recording conditions |
| MSU-MFSD | 2015 | Wen et al. (2015) | 110 | 330 | 55 | 3 | |
| MSU-USSA | 2016 | Patel et al. (2016) | 1140 | 9120 | 1140 | 2 | Multiple devices for replay spoofing; |
| MLFP | 2017 | Agarwal et al. (2017) | 150 | 1200 | 10 | 2 | Visible, near infra-red and thermal modes for each sample |
| Oulu-NPU | 2017 | Boulkenafet et al. (2017) | 990 | 3960 | 55 | 4 | Varied environments |
| SiW | 2018 | Liu et al. (2018a) | 1320 | 3300 | 165 | 6 | Varying subject movements, camera angles and facial expressions |
| ROSE-Youtu | 2018 | Li et al. (2018) | 500 | 2850 | 20 | 3 | |
| SiW-M | 2019 | Liu et al. (2019c) | 660 | 968 | 493 | 13 | Different scenarios (varying movement, light, camera quality, distance to camera) |
| HQ-WMCA | 2020 | Mostaani et al. (2020) | 555 | 2349 | 51 | 10 | Focus on varied attacks and a multi-modal character |
| DMAD | 2020 | Wang et al. (2020b) | 900 | 1800 | 300 | 6 | |
| WFAS | 2023 | Wang et al. (2023) | 529.571 | 853.729 | 469.920 | 17 | Images scraped from the internet and grouped with face recognition |

Tabela 3.1: Studied datasets' main characteristics.

This work's evaluation benchmarks rely on the CASIA-FASD (Zhang et al., 2012), Replay-Attack (Chingovska et al., 2012), MSU-MFSD (Wen et al., 2015), Oulu-NPU (Boulkenafet et al., 2017), SiW (Liu et al., 2018a) and Rose-Youtu (Li et al., 2018) datasets. While individually some of them may not represent a challenge to recent state-of-the-art FAS methods, when combined into cross-database scenarios they become an open problem to the community. A

particular characteristic of MSU-MFSD and Oulu-NPU is that they embrace mobile usage as a primary focus and so try to approximate mobile phone usage in recording live samples. We use the four intra-datasets proposed by Boulkenafet et al. (2017) with the Oulu-NPU dataset, as well as six domain generalization benchmarks using the four aforementioned datasets. They are of two types: *standard* DG benchmarks, which involve three datasets for training and another for testing, and *limited source domain* DG benchmarks, which involve just two datasets for training and one for testing. These benchmarks as well as acronyms used for them in this work are presented in Table 3.2. Besides these benchmarks, new ones are proposed in this work. They are presented in later chapters.

| Name | Type | Train datasets | Test Dataset |
|---|---|---|---|
| ICM to O | Standard | Replay-Attack, CASIA-FASD, MSU-MFSD | Oulu-NPU |
| OCI to M | Standard | Oulu-NPU, Replay-Attack, CASIA-FASD | MSU-MFSD |
| OMI to C | Standard | Oulu-NPU, MSU-MFSD, Replay-Attack | CASIA-FASD |
| OCM to I | Standard | Oulu-NPU, CASIA-FASD, MSU-MFSD | Replay-Attack |
| MI to C | Limited source domain | MSU-MFSD, Replay-Attack | CASIA-FASD |
| MI to O | Limited source domain | MSU-MFSD, Replay-Attack | Oulu-NPU |

Tabela 3.2: Description of DG benchmarks used in this work. The first letters are the initials of each dataset in the train set, and the last letter is the initial of the test dataset. *I* corresponds to Replay-Attack (and is often replaced with an *R*), and is used because Replay-Attack comes from the Idiap Research Institute (Chingovska et al., 2012). Often these benchmarks are referred to as three or four letters, following the convention of each letter representing a dataset and the last letter representing the test dataset (so ICM to O could also be referred to as ICMO, CIMO, MICO, MRCO, CRMO and RCMO, for example), and train set letters are often swapped.

The Oulu-NPU benchmarks have slight different intentions in their designs and allow for a reasonable understanding of a model's pitfalls. Benchmark I is designed to evaluate a method's generalization to previously unseen environmental conditions (illumination and background scene). Benchmark II evaluates the method's capability to perform well against previously unseen attack instruments, and so the test set consists of attacks created with new artifacts. Benchmark III evaluates sensor interoperability and consists of a Leave-One-Camera-Out strategy in which the test set consists of data collected from a camera not present in the train set. Benchmark IV evaluates everything from the first three benchmarks at the same time, i.e., a model's capability to generalize to new environments, attacks and input sensors.

Figures 3.3 and 3.4 show example live and spoof samples, respectively, from each of these four datasets, as well as their corresponding depth maps.

## 3.2 METHODS FOR FACE ANTI-SPOOFING

To have an overview of the history and state of methods for FAS, it is useful to use as reference the taxonomy presented in Chapter 2, namely the one in Figure 2.2. In the following explanation, models are grouped by their respective categories, starting with hybrid approaches and moving on to traditional DL methods (divided between BCE-exclusive and pixelwise-supervised models) and generalized DL methods (divided between Domain Generalization and the rest - Domain Adaptation, zero- and few-shot learning and anomaly detection).

Hybrid approaches are the most classical. Even though today there are no state-of-the-art methods belonging to this category, expert knowledge is still useful and guides much of the design choices for new approaches. Tan et al. (2010) use a lambertian model with two strategies for obtaining latent features. Chingovska et al. (2012) base their work on Local Binary Patterns and their variations. Wen et al. (2015) extract a feature vector from specular reflection, blurriness,

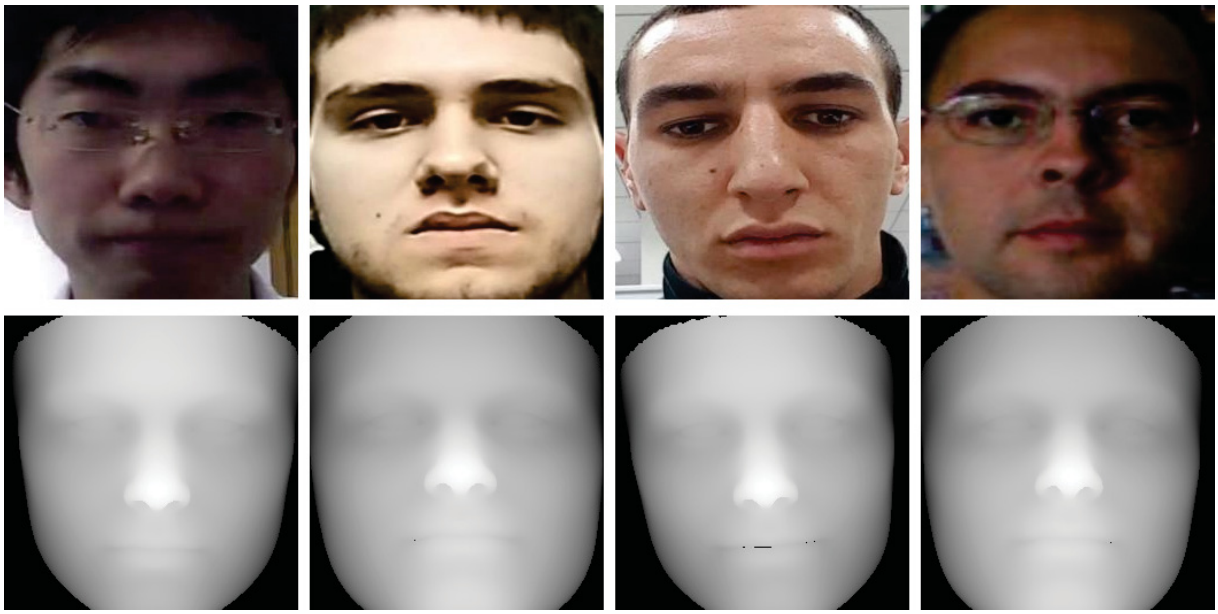


Figura 3.3: Example bona fide samples from each of CASIA-FASD, MSU-MFSD, Oulu-NPU and Replay-Attack, in this order. The images from the second row correspond to the depth map generated from the images right above them. Sources: Zhang et al. (2012); Wen et al. (2015); Boulkenafet et al. (2017); Chingovska et al. (2012).

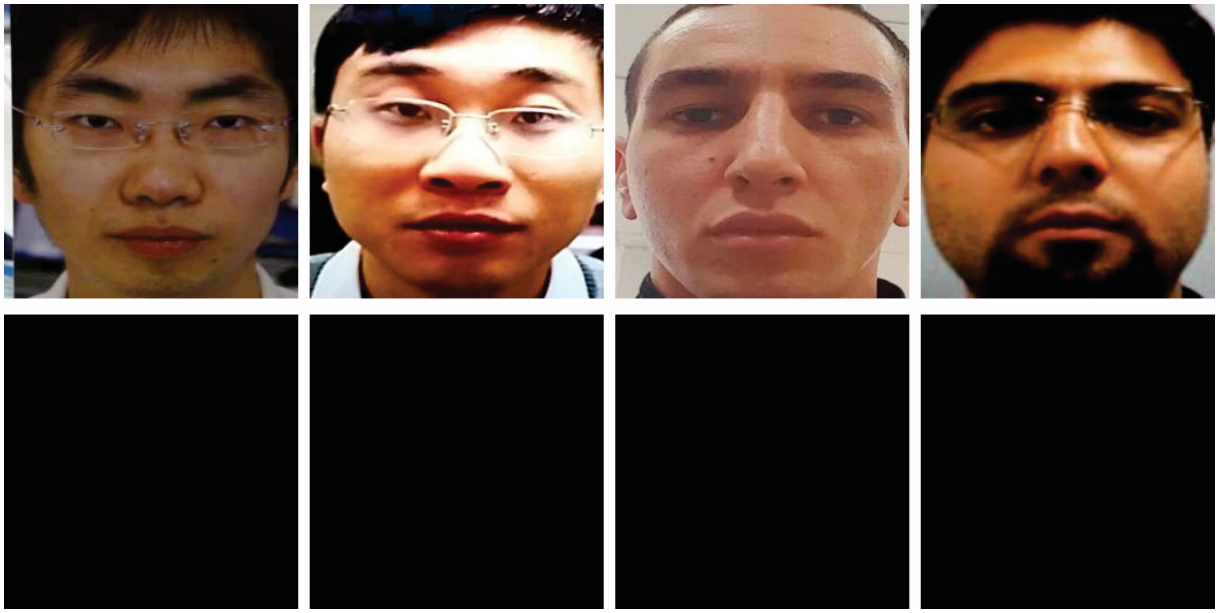


Figura 3.4: Example attack samples from each of CASIA-FASD, MSU-MFSD, Oulu-NPU and Replay-Attack, in this order. The images from the second row correspond to the depth map generated from the images right above them, but note that in the case of spoofs the depth map is always an empty one. Sources: Zhang et al. (2012); Wen et al. (2015); Boulkenafet et al. (2017); Chingovska et al. (2012).

chromatic moment and color intensity to train an ensemble of Support Vector Machines (SVM). Patel et al. (2016) build on top of this approach by using surface reflection, moire patterns, color distortion and shape deformation. Boulkenafet et al. (2016) use color textures on top of luminance information. These methods are precursors to more recent ones that use specific features together with more robust CNN models: Koshy e Mahmood (2019) use texture analysis and Li et al. (2019) use motion blur. Other work may be considered subject-oriented in that they use features related to the input subject: Killioğlu et al. (2017) use eye pupil movement, Chan et al. (2018)

use two images where one is illuminated by a camera flash, Singh e Arora (2018) use eye blink and lip and chin movement, and Liu et al. (2019b) use a kinect device for input. Luo et al. (2018) use multi-level face region cropping together with LSTMs to avoid losing valuable information from the background. Chen et al. (2020) use an illumination-invariant multi-scale retnex space.

In current days it is easy to foresee that the next step in the history of FAS models is the dominance of deep learning and BCE-based models. Yang et al. (2015) integrate the spoofing step with the face identification step to leverage knowledge of the user database, but fail to do so in a scalable manner. Some early methods were based on extracting features with a CNN and then feeding those to an SVM (Li et al., 2016; Ito et al., 2017). Many different architectures were explored over the years for the task of FAS, for example Deep Tree Networks (Liu et al., 2019c), LSTMs Yang et al. (2019), multi-stream networks Shen et al. (2019) and deep belief networks Garg et al. (2020). Wang et al. (2019) present a multi-modal approach with four branches for RGB, depth, infrared and a concatenation of all three, which may be more efficient at the cost of practicality. Liu et al. (2021a) present an RGB-based approach that is extendable when multi-modal data is available. Sanghvi et al. (2021) use three subnetworks to detect different types of attacks in each one.

One thing that has been often more effective than straightforward deep learning with BCE is the usage of pixelwise supervision, often in the form of pseudo depth maps (Atoum et al., 2017; Wang et al., 2020b; Zhang et al., 2020; Yu et al., 2020b, 2021; Zheng et al., 2021; Wang et al., 2021b). Atoum et al. (2017) use depth map supervision in training a two-stream CNN. Jourabloo et al. (2018) trains a network to remove spoof noise from images. Liu et al. (2018a) use remote photoplethysmography signals on top of the RGB input. Wang et al. (2020b) mix depth and movement in pattern recognition. Zhang et al. (2020) and Wang et al. (2021b) use depth supervision in an adversarial scheme, with the latter work using the depth estimation network as a feature extractor for a classifier network. Deb e Jain (2021) employ self-supervision in learning local discriminative cues with results comparable to the state of the art, but hindered by low availability of data. Other works use depth supervision with central difference convolutions (Yu et al., 2020b, 2021). Zheng et al. (2021) use both depth and multi-scale information.

As intra-dataset benchmarks became less of a challenge to the state-of-the-art, generalization became more important. As illustrated in Figure 2.2, we divided generalization models between DG and others. In domain adaptation, Li et al. (2018) aim to learn the classifier by encountering a space where domain distribution similarity can be measured and optimized. In the same subtask, Wang et al. (2021a) apply an unsupervised model adaptor. George e Marcel (2021) fine-tune a vision transformer for transfer learning in zero-shot FAS. Quan et al. (2021) present a few-shot method that requires as few as 50 training samples to be competitive with the state of the art.

For domain generalization, Shao et al. (2019) propose a multi-adversarial method with auxiliary depth. Focusing on representation learning, Wang et al. (2020a) employ disentangled representation learning and Wang et al. (2022) split the input image into content and style features. Other works focus on objective engineering: Jia et al. (2020) develop an end-to-end DG framework that leverages asymmetric triplet mining, Sun et al. (2023) embrace domain differences by instead encouraging separability and alignment in embeddings and Le e Woo (2024) design convergence on an optimal flat minimum that responds well to domain shifts. Some models benefit from other modalities, with highlights being Heusch et al. (2020), that use shortwave infrared to improve the generalization capability of CNN-based models, and Liu et al. (2021b), that fuse features from different modalities to improve feature representation. Chen et al. (2021) extract features from both a high-frequency domain and an enhanced image to achieve generalization, and Zhou et al. (2023) instead whiten an instance's domain-variant features.

Liu et al. (2019a) and Purnapatra et al. (2021) both present reflections on challenges for researchers in DG-FAS. Authors of the first challenge argued that FAS remains challenging due to lack of generalization because methods often rely too heavily on known data and may be found fragile once confronted with unknown acquistion devices, attack methods and spoofing mediums. They also point out that the ubiquitous usage of softmax loss might lead models to value arbitrary cues, which is an argument in favour of pixelwise supervision. In the more recent challenge, it is observed that general performance worsened in comparison to previous editions, which is mainly attributed to increased complexity in testing data.

| Method | Year | OCI to M | | OMI to C | | OCM to I | | ICM to O | | MI to C | | MI to O | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HTER% ↓ | AUC% ↑ | HTER% ↓ | AUC% ↑ | HTER% ↓ | AUC% ↑ | HTER% ↓ | AUC% ↑ | HTER% ↓ | AUC% ↑ | HTER% ↓ | AUC% ↑ |
| MADDG (Shao et al., 2019) | 2019 | 17.69 | 88.06 | 24.5 | 84.51 | 22.19 | 84.99 | 27.98 | 80.02 | 41.02 | 64.33 | 39.35 | 65.10 |
| SSDG-M (Jia et al., 2020) | 2020 | 16.67 | 90.47 | 23.11 | 85.45 | 18.21 | 94.61 | 25.17 | 81.83 | 31.89 | 71.29 | 36.01 | 66.88 |
| SSDG-R (Jia et al., 2020) | 2020 | 7.38 | 97.17 | 10.44 | 95.94 | 11.71 | 96.59 | 15.61 | 91.54 | | | | |
| Wang et al. (2020a) | 2020 | 17.02 | 90.10 | 19.68 | 87.43 | 20.87 | 86.72 | 25.02 | 81.47 | 31.67 | 75.23 | 34.02 | 72.65 |
| Wang et al. (2021a) | 2021 | 15.4 | 91.8 | 24.5 | 84.4 | 15.6 | 90.1 | 23.1 | 84.3 | | | | |
| S-CNN+PL+TC+AT (Quan et al., 2021) | 2021 | 7.82 ± 1.21 | 97.67 ± 1.09 | **4.01 ± 0.81** | **98.96 ± 0.77** | 10.36 ± 1.86 | 97.16 ± 1.04 | 14.23 ± 0.98 | 93.66 ± 0.75 | | | | |
| DFA (Wang et al., 2021b) | 2021 | 19.40 | 86.87 | 22.03 | 87.71 | 21.43 | 88.81 | 18.26 | 89.40 | | | | |
| Chen et al. (2021) | 2021 | | | | | | | | | 32.3 | | | |
| MA-Net (Liu et al., 2021a) | 2021 | 20.8 | | 25.6 | | 24.7 | | 26.3 | | | | | |
| SSAN-M (Wang et al., 2022) | 2022 | 10.42 | 94.76 | 16.47 | 90.81 | 14.00 | 94.58 | 19.51 | 88.17 | 30.00 | 76.20 | 29.44 | 76.62 |
| SSAN-R (Wang et al., 2022) | 2022 | 6.67 | **98.75** | 10.00 | 96.67 | 8.88 | 96.79 | 13.72 | 93.63 | | | | |
| IADG (Zhou et al., 2023) | 2023 | 5.41 | 98.19 | 8.70 | 96.44 | 10.62 | 94.50 | 8.86 | 97.14 | 24.07 | 85.13 | 18.47 | **90.49** |
| SA-FAS (Sun et al., 2023) | 2023 | 5.95 | 96.55 | 8.78 | 95.37 | 6.58 | 97.54 | 10.00 | 96.23 | | | | |
| GAC-FAS (Le e Woo, 2024) | 2024 | **5.00** | 97.56 | 8.20 | 95.16 | **4.29** | **98.87** | **8.60** | **97.16** | **16.91** | **88.12** | **17.88** | 89.67 |

Tabela 3.3: Reported performances (HTER% and AUC%) of studied methods in DG benchmarks. Shown values are as reported by the authors. The best value in each column is highlighted, with the best HTER rate being the one with the smallest value (small error) and the best AUC value being the largest. Works are sorted by year, and the results of S-CNN+PL+TC+AT (Quan et al., 2021) are reported with mean and standard error values due to the particular training of the model (with only 50 samples).

Table 3.3 presents a comparison of results for studied methods in domain generalization, namely in the main cross-dataset scenarios involving CASIA-FASD, Replay-Attack, MSU-MFSD and Oulu-NPU (Zhang et al., 2012; Chingovska et al., 2012; Wen et al., 2015; Boulkenafet et al., 2017). We show these benchmarks specifically due to their relevance to the present work. Tables 3.4, 3.5, 3.6 and 3.7 present a comparison of results for studied methods in Oulu-NPU intra-dataset benchmarks 1 through 4.

| Method | APCER% ↓ | BPCER% ↓ | ACER% ↓ |
|---|---|---|---|
| S-CNN+PL+TC (Quan et al., 2021) | 0.6 ± 0.4 | **0.0 ± 0.0** | 0.4 ± 0.2 |
| DFA (Wang et al., 2021b) | 0.87 | 1.06 | 0.92 |
| Chen et al. (2021) | 3.8 | 2.9 | 3.4 |
| MSNet (Zheng et al., 2021) | 1.4 | 1.8 | 1.6 |
| CDCN (Yu et al., 2020a) | **0.4** | 1.7 | 1.0 |
| CDCN++ (Yu et al., 2020a) | **0.4** | **0.0** | **0.2** |
| SSR-FCN (Deb e Jain, 2021) | 1.5 | 7.7 | 4.6 |
| Zhang et al. (2020) | 1.7 | 0.8 | 1.3 |
| FAS-SGTD (Wang et al., 2020b) | 2.0 | **0.0** | 1.0 |
| STASN (Yang et al., 2019) | 1.2 | 0.8 | 1.0 |
| TSCNN (Chen et al., 2020) | 5.1 | 6.7 | 5.9 |
| Jourabloo et al. (2018) | 1.2 | 1.7 | 1.5 |
| CNN-RNN (Liu et al., 2018a) | 1.6 | 1.6 | 1.6 |
| DC-CDN (Yu et al., 2021) | 0.5 | 0.3 | 0.4 |

Tabela 3.4: Reported APCER%, BPCER% and ACER% values of studied methods in Oulu-NPU benchmark 1. Shown values are as reported by the authors. The best value in each column is highlighted, with the best error rates being the lowest. Works are sorted by year, and the results of S-CNN+PL+TC+AT (Quan et al., 2021) are reported with mean and standard error values due to the particular training of the model (with only 50 samples).

## 3.3 KEY PROBLEMS

Currently, one main problem can be identified for FAS: increasing generalization capability in classification methods. This applies to multiple perspectives of domains, such as acquisition device, attack types and environmental conditions. As this is an adversarial

| Method | APCER% ↓ | BPCER% ↓ | ACER% ↓ |
|---|---|---|---|
| S-CNN+PL+TC (Quan et al., 2021) | 1.7 ± 0.9 | **0.6 ± 0.3** | 1.2 ± 0.5 |
| DFA (Wang et al., 2021b) | 3.84 | 2.11 | 2.88 |
| Chen et al. (2021) | 3.6 | 1.2 | 2.4 |
| MSNet (Zheng et al., 2021) | 2.6 | 0.8 | 1.7 |
| CDCN (Yu et al., 2020a) | 1.5 | 1.4 | 1.5 |
| CDCN++ (Yu et al., 2020a) | 1.8 | 0.8 | 1.3 |
| SSR-FCN (Deb e Jain, 2021) | 3.1 | 3.7 | 3.4 |
| Zhang et al. (2020) | 1.1 | 3.6 | 2.4 |
| FAS-SGTD (Wang et al., 2020b) | 2.5 | 1.3 | 1.9 |
| STASN (Yang et al., 2019) | 1.4 | 0.8 | **1.1** |
| TSCNN (Chen et al., 2020) | 7.6 | 2.2 | 4.9 |
| Jourabloo et al. (2018) | 4.2 | 4.4 | 4.3 |
| CNN-RNN (Liu et al., 2018a) | 2.7 | 2.7 | 2.7 |
| DC-CDN (Yu et al., 2021) | **0.7** | 1.9 | 1.3 |

Tabela 3.5: Reported APCER%, BPCER% and ACER% values of studied methods in Oulu-NPU benchmark 2. Shown values are as reported by the authors. The best value in each column is highlighted, with the best error rates being the lowest. Works are sorted by year, and the results of S-CNN+PL+TC+AT (Quan et al., 2021) are reported with mean and standard error values due to the particular training of the model (with only 50 samples).

| Method | APCER% ↓ | BPCER% ↓ | ACER% ↓ |
|---|---|---|---|
| S-CNN+PL+TC (Quan et al., 2021) | 1.5 ± 0.9 | 2.2 ± 1.0 | **1.7 ± 0.8** |
| DFA (Wang et al., 2021b) | 1.9 ± 1.6 | 3.8 ± 6.4 | 2.8 ± 2.7 |
| Chen et al. (2021) | 3.8 ± 1.3 | **1.1 ± 1.1** | 2.5 ± 0.8 |
| MSNet (Zheng et al., 2021) | 2.0 ± 2.6 | 3.9 ± 2.2 | 2.8 ± 2.4 |
| CDCN (Yu et al., 2020a) | 2.4 ± 1.3 | 2.2 ± 2.0 | 2.3 ± 1.4 |
| CDCN++ (Yu et al., 2020a) | 1.7 ± 1.5 | 2.0 ± 1.2 | 1.8 ± 0.7 |
| SSR-FCN (Deb e Jain, 2021) | 2.9 ± 2.1 | 2.7 ± 3.2 | 2.8 ± 2.2 |
| Zhang et al. (2020) | 2.8 ± 2.2 | 1.7 ± 2.6 | 2.2 ± 2.2 |
| FAS-SGTD (Wang et al., 2020b) | 3.2 ± 2.0 | 2.2 ± 1.4 | 2.7 ± 0.6 |
| STASN (Yang et al., 2019) | **1.4 ± 1.4** | 3.6 ± 4.6 | 2.5 ± 2.2 |
| TSCNN (Chen et al., 2020) | 3.9 ± 2.8 | 7.3 ± 1.1 | 5.6 ± 1.6 |
| Jourabloo et al. (2018) | 4.0 ± 1.8 | 3.8 ± 1.2 | 3.6 ± 1.6 |
| CNN-RNN (Liu et al., 2018a) | 2.7 ± 1.3 | 3.1 ± 1.7 | 2.9 ± 1.5 |
| DC-CDN (Yu et al., 2021) | 2.2 ± 2.8 | 1.6 ± 2.1 | 1.9 ± 1.1 |

Tabela 3.6: Reported APCER%, BPCER% and ACER% values of studied methods in Oulu-NPU benchmark 3. Shown values are as reported by the authors. The best value in each column is highlighted, with the best error rates being the lowest. Works are sorted by year.

| Method | APCER% ↓ | BPCER% ↓ | ACER% ↓ |
|---|---|---|---|
| S-CNN+PL+TC (Quan et al., 2021) | 5.2 ± 2.0 | 4.6 ± 4.1 | 4.8 ± 2.0 |
| DFA (Wang et al., 2021b) | 4.0 ± 4.1 | 3.0 ± 4.9 | 3.5 ± 2.4 |
| Chen et al. (2021) | 5.9 ± 3.3 | 6.3 ± 4.7 | 6.1 ± 4.1 |
| MSNet (Zheng et al., 2021) | 4.2 ± 5.2 | 4.6 ± 3.8 | 4.4 ± 4.5 |
| CDCN (Yu et al., 2020a) | 4.6 ± 4.6 | 9.2 ± 8.0 | 6.9 ± 2.9 |
| CDCN++ (Yu et al., 2020a) | 4.2 ± 3.4 | 5.8 ± 4.9 | 5.0 ± 2.9 |
| SSR-FCN (Deb e Jain, 2021) | 8.3 ± 6.8 | 13.3 ± 8.7 | 10.8 ± 5.1 |
| Zhang et al. (2020) | 5.4 ± 2.9 | 3.3 ± 6.0 | 4.4 ± 3.0 |
| FAS-SGTD (Wang et al., 2020b) | 6.7 ± 7.5 | 3.3 ± 4.1 | 5.0 ± 2.2 |
| STASN (Yang et al., 2019) | **0.9 ± 1.8** | 4.2 ± 5.3 | **2.6 ± 2.8** |
| TSCNN (Chen et al., 2020) | 11.3 ± 3.9 | 9.7 ± 4.8 | 9.8 ± 4.2 |
| Jourabloo et al. (2018) | 5.1 ± 6.3 | 6.1 ± 5.1 | 5.6 ± 5.7 |
| CNN-RNN (Liu et al., 2018a) | 9.3 ± 5.6 | 10.4 ± 6.0 | 9.5 ± 6.0 |
| DC-CDN (Yu et al., 2021) | 5.4 ± 3.3 | **2.5 ± 4.2** | 4.0 ± 3.1 |

Tabela 3.7: Reported APCER%, BPCER% and ACER% values of studied methods in Oulu-NPU benchmark 4. Shown values are as reported by the authors. The best value in each column is highlighted, with the best error rates being the lowest. Works are sorted by year.

task, spoofing techniques develop as fast as methods used to prevent spoofing, and so domain generalization becomes even more difficult.

Besides that, there is the issue of data availability. Current state-of-the-art FAS models rely heavily on large amounts of sensitive, high-quality data, which is very difficult to acquire for technical, practical and legal reasons. Very little research has been invested in tackling this issue, which could be seen as putting out a fire with a glass of water. Data augmentation, for example, is a pathway for enriching training datasets, not explored in most FAS research.

Finally, it is preliminary to take note of the fact that the benchmarks behind DG-FAS models are fundamentally biased, which overall creates noise in the data behind the collective decision-making that guides the direction of DG-FS research. This is particularly problematic when one considers the importance of generalization capability in real-world models.

## 3.4 CONCLUDING REMARKS

Although some of the mentioned models and datasets are revisited in later chapters for result reproduction and new experiments, the main idea of this chapter is to paint the picture of how FAS arrived where it is today. For that, methods and datasets were presented and discussed. This work builds upon the idea of Patch Exchange (Yu et al., 2021) to further explore model performance enhancement through exchange augmentation strategies, and on the state-of-the-art DG-FAS benchmarks to explore unbiased benchmark possibilities. This is pursued in a model-agnostic manner and a pool of methods and datasets is experimented with, as described in Chapter 5.

# 4 PROPOSED APPROACHES FOR ENHANCED MODEL PERFORMANCE AND APPLICABILITY

This chapter explains the proposed approach for data augmentation in FAS, which is based on Patch Exchange (PE) (Yu et al., 2021), and the proposed unbiased benchmarks for DG-FAS. For both proposals, the motivating factors and strategies themselves are explained. Before entering these matters, baseline methods to be used in experiments are also presented in detail.

## 4.1 BASELINE METHODS

We now present four baseline methods to be used in this work for experiments and methodological considerations. Three of them are domain generalization works (all except DC-CDN). In particular, one of the methods (DC-CDN) includes a data augmentation strategy, Patch Exchange, which we build upon in this work. The study of data augmentation strategies for FAS, the main theme in this work, is not a common research topic, and to the best of our knowledge, the Patch Exchange method is the first strategy developed with FAS in mind.

### 4.1.1 Gradient Alignment for Cross-Domain FAS (GAC-FAS)

The goal behind GAC-FAS is to model an objective function that encourages the model to converge towards a flat minimum, which is shown to be advantageous for DG and is not common in most FAS works prior to this. This is achieved without additional learning modules. GAC-FAS identifies ascending points for each domain and regulates the generalization gradient updates at these points with empirical risk minimization. The obtained flat minimum makes for models that are robust against domain shifts, and the authors report state-of-the-art results (Le e Woo, 2024).

### 4.1.2 Shuffled-Style Assembly Network (SSAN)

The main idea behind SSAN (Wang et al., 2022) is to have a two-branch network process style and content features (which will be explained next) separately. This was the first work to do this with shuffled assembly.

**Content** features are extracted with Batch Normalization and mostly related to global image statistics such as semantic features and physical attributes. **Style** features are extracted with Instance Normalization and focus on local characteristics such as liveness-related texture and domain-specific external factors. It is relevant to take a step back in analyzing the normalization structures utilized here, which is done in the next subsection.

#### *4.1.2.1 Normalization options*

We now briefly go over Batch and Instance Normalization in the context of style transfer, and present two alternatives to Instance Normalization (Huang e Belongie, 2017). Below, $\mu, \sigma$ represent the mean and standard deviation functions, respectively.

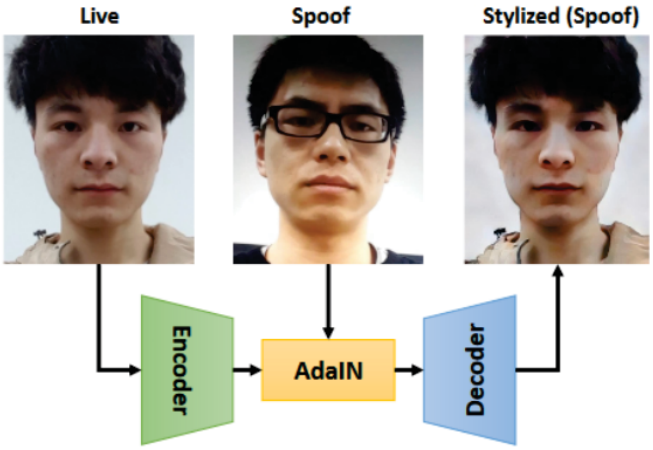



Figura 4.1: Spoof stylization with AdaIN (Huang e Belongie, 2017). Source: Wang et al. (2022).

**Batch Normalization** The Batch Normalization (BN) operation is calculated as follows

$$BN(x) = \gamma \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta, \tag{4.1}$$

where $\gamma, \beta \in \mathbb{R}^C$ are affine parameters learned from data and $\mu, \sigma$ are calculated across batch size and spatial dimensions independently for each feature channel (i.e., one mean for each channel).

**Instance Normalization** Very similarly, the Instance Normalization (IN) is calculated as follows

$$IN(x) = \gamma \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta. \tag{4.2}$$

The only difference from BN is that $\mu$ and $\sigma$ are calculated across spatial dimensions independently for each channel **and** each sample (i.e., one mean for each pair of channel and sample).

It is argued by Huang e Belongie (2017) that IN performs a form of style normalization by normalizing feature statistics (mean and variance). It follows from this argument that BN is less effective in style transfer because it would perform this style normalization among all samples in a batch.

**Conditional Instance Normalization** Instead of learning a single set of affine parameters $\gamma, \beta$, Dumoulin et al. (2017) propose CIN to learn a different set of parameters for each style $s$:

$$CIN(x; s) = \gamma^s \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta^s. \tag{4.3}$$

This is calculated similarly to IN. It requires a lot more parameters, and in order to adapt the model to a new style, it is necessary to retrain the network.

**Adaptive Instance Normalization** With the argument of style normalization in mind, AdaIN (Huang e Belongie, 2017) is proposed as follows to adapt an input $x$ to the style of another input $y$:

$$AdaIN(x, y) = \sigma(y) \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y)$$

This is calculated similarly to IN. Note that there are no learnable affine parameters. Figure 4.1 illustrates the effects of AdaIN in practice.

#### *4.1.2.2 Network Overview*

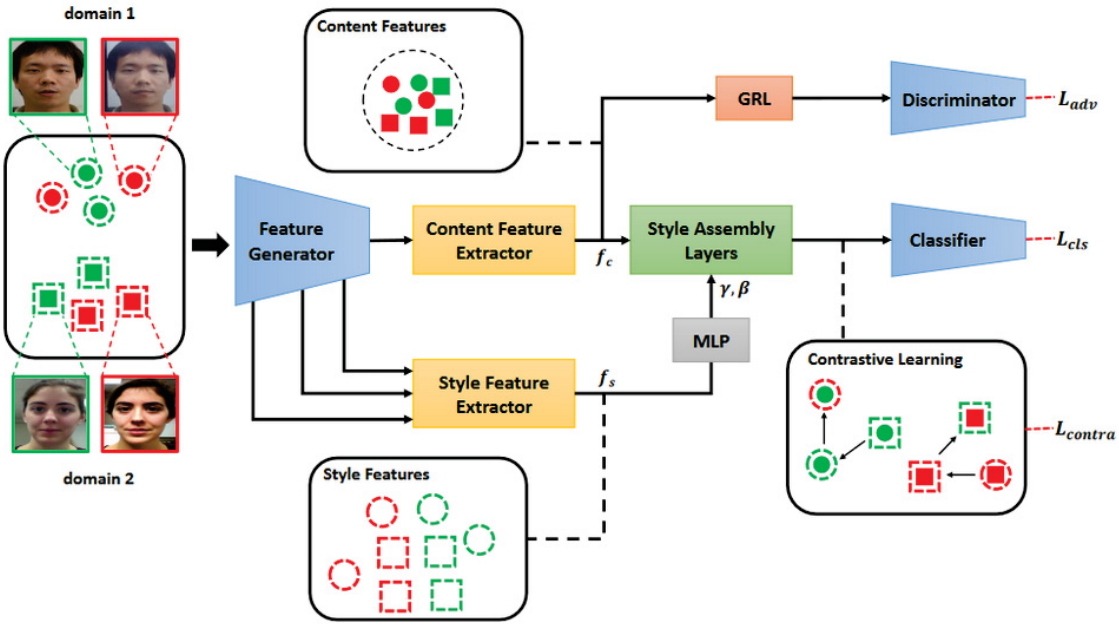



Figura 4.2: Overview of SSAN. Source: Wang et al. (2022).

We now present an overview of the SSAN architecture. First, a **Feature Generator** receives as input data with domain and live/spoof labels. This generator is a shallow network that captures multi-scale, low-level information. Its output is fed into two feature extractors: one for content and one for style features.

**Content Feature Extractor:** based on the hypothesis that small distribution discrepancies will exist between domains (since they share a lot of similarities i.e. presence of faces, shape and sizes), this module is designed to produce features that are indistinguishable from one domain to another. Its output is fed into a Domain Discriminator and the Style Assembly Layers.

**Domain Discriminator:** this network learns to distinguish features between domains and is trained in an adversarial manner - BCE loss (with respect to the domain labels) is iteratively minimized by the Discriminator and maximized by the Content Feature Extractor. A Gradient Reversal Layer (GRL) is used between these modules to reverse gradients during backward passes.

**Style Feature Extractor:** this network uses multi-scale features from the generator, since style characteristics have different scales (e.g., brightness is broad-scale, texture is local-scale). Features from this extractor should be separated by class, regardless of domain. Its output is fed into a Multi-Layer Perceptron and then into the Style Assembly Layers.

**Style Assembly Layers:** these layers assemble content and style features from the same input sample into embeddings used for classification. The classifier network then produces either a binary label (supervised with BCE in SSAN-R) or a pseudo depth map (supervised with MSE in SSAN-M). These two possibilities of SSAN (SSAN-R and SSAN-M) are different variations of the network.

Another thing that Style Assembly Layers do is shuffle style and content features from different samples (shuffled-style assembly). This is done with a contrastive learning scheme to emphasize and suppress liveness-related and domain-related features, respectively.

**Contrastive learning:** the idea in using contrastive learning in this network is to enforce that instances mapped to close regions in the feature space have the same source class in both their content and style features, i.e., that there is no trace of spoofing in either feature source.

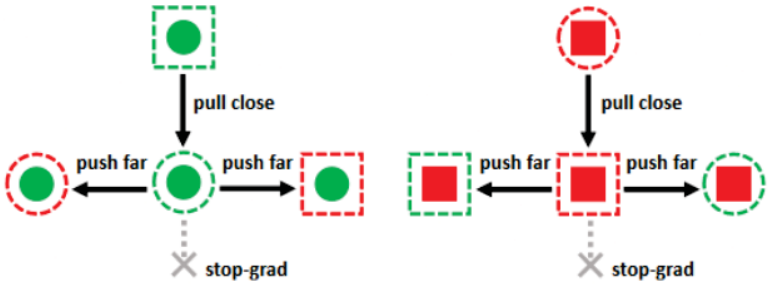



Figura 4.3: Contrastive learning illustration. Source: Wang et al. (2022).

### 4.1.3 Instance-Aware Domain Generalization (IADG)

IADG removes the need for domain labels by learning to whiten an instance's domain-invariant features in a domain-agnostic manner.

The work building up to IADG (Zhou et al., 2023) starts off with the key observation that the approach often taken by DG methods is fundamentally flawed. What the authors argue is that learning a domain-oriented representation for samples is not enough the be able to generalize to other domains, because what we currently use as domain labels is some coarse-grained information that could make the model ill-guided in the task. Consider for example illumination variations. This is a simple example of domain characteristics. What can happen is that one of the datasets used for training has very distinct illuminations characteristics in comparison with the other train datasets, and the model might learn this arbitrary cue to perform well in the DG auxiliary task. Two other issues are that these labels cannot comprehensively reflect the real domain distributions and only focus on the perspective of domain distributions, which means the model might still be sensitive to unfaithful instance style information.

Based on that, the authors propose an Instance-Aware Domain Generalization method to obtain style-insensitive features that are instance-aligned instead of domain-aligned (which, as argued above, is a much more fine-grained strategy for alignment - see Figure 4.4). This method learns to embed features in such a way that obtained embeddings are insensitive to instance-specific styles. This is achieved with a few key components described below and visualized in Figure 4.5.

#### *4.1.3.1 Dynamic Kernel Generator (DKG)*

In the multi-domain scenario, samples will have very varied features, which makes it an inefficient strategy to rely on a single set of network convolution weights to embed features (Zhou et al., 2023). DKG is a way to solve this issue by learning to generate a different kernel to process each sample. In practice, each input feature is split into two halves: the first goes through static convolution filters, and the second goes through dynamic filters. The kernel of these dynamic filters is obtained through global average pooling and convolution of the second

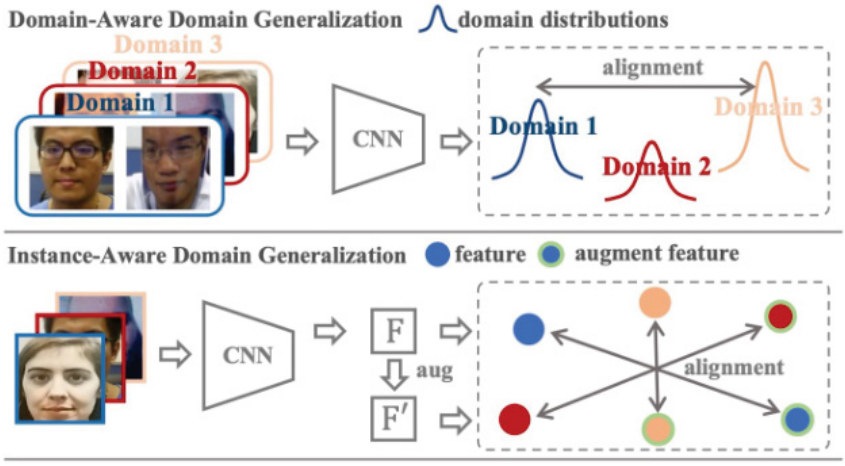



Figura 4.4: Illustration of conventional and IADG-proposed alignment with respect to domain characteristics. Source: Zhou et al. (2023).

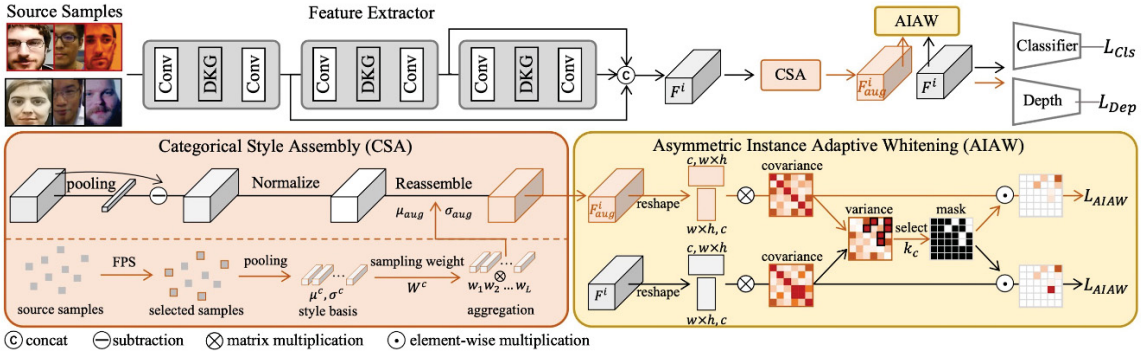



Figura 4.5: IADG network overview. Source: Zhou et al. (2023).

half of the input features. Both processed halves are concatenated and once again convoluted against a static kernel before finally being output by the DKG. This module enhances the learning of instance-specific features that contribute to domain generalization.

### *4.1.3.2 Categorial Style Assembly (CSA)*

The CSA module helps obtain augmented features by mixing style from different samples, similar to what is done in SSAN (Wang et al., 2022). It consists of, given a set of same-class samples, selecting some of the most distant from the rest (inspired by Farthest Point Sampling) to be used in augmenting the entire set.

From these selected samples, style information $\mu_{base}$ and $\sigma_{base}$ is extracted and then weighted with weights $W$: $\mu_{aug} = W \cdot \mu_{base}$ and $\sigma_{aug} = W \cdot \sigma_{base}$. The value of $W$ is obtained from a Dirichlet distribution $B([\alpha_1, \cdots, \alpha_L])$, where all concentration parameters $\alpha_i$ are set to $\frac{1}{L}$ (this is a conservative choice resulting from our lack of *a priori* knowledge about each style's importance). Finally, each augmented feature map $F_{aug}$ is obtained from an input $F_{org}$ with AdaIN (Huang e Belongie, 2017): $F_{aug} = \sigma_{aug}\left(\frac{F_{org} - \mu(F_{org})}{\sigma(F_{org})}\right) + \mu_{aug}$.

This process is executed separately for each class (real and spoof) to simplify label handling and create more realistic augmented features.

#### *4.1.3.3 Asymmetric Instance Adaptive Whitening (AIAW)*

Feature covariance matrices have been shown to reveal domain-specific features in the tasks of image translation (Cho et al., 2019), style transfer (Li et al., 2017) and domain adaptation (Roy et al., 2020; Sun e Saenko, 2016), which is a great hint in the direction of doing the same thing for DG FAS. The difference between these tasks and FAS that makes this a difficult problem is that in the case of FAS we are not centered on domain generalization, but instead use it as an auxiliary direction to detect face liveness: if we simply removed domain-specific features with instance whitening we would also remove domain-invariant features which are discriminative for classification. So the goal of the Asymmetric Instance Adaptive Whitening module is to find a way to suppress and highlight the correct covariance.

This module receives as input both an original feature map $F_{org}$ and its augmented (from CSA) counterpart $F_{aug}$. Each of these feature maps goes through a normalization step and its covariance matrix ($\Sigma_F = F \cdot F^T$) is computed. The variance $\sigma^2_\Sigma$ between $\Sigma_{F_{org}}$ and $\Sigma_{F_{aug}}$ is calculated as follows:

$$\mu_\Sigma = \frac{1}{2}\left(\Sigma_{F_{org}} + \Sigma_{F_{aug}}\right)$$

$$\sigma^2_\Sigma = \frac{1}{2}\left(\left(\Sigma_{F_{org}} - \mu_\Sigma\right)^2 + \left(\Sigma_{F_{aug}} - \mu_\Sigma\right)^2\right)$$

Having calculated that for each feature map $F$, it is possible to calculate the mean variance for all samples as $V = \frac{1}{N}\sum \sigma^2_\Sigma$, where $N$ is the number of samples. We can then seek to whiten the $K$ largest values in $V$, which is done through auxiliary supervision. This whitening is bilateral, in that both the original and the augmented features are whitened, in order to further guarantee that features will have small variations corresponding to style changes.

In order to have bona fide features form a compact cluster while spoof features are spread in the feature space, the authors introduce the concept of asymmetry, and use different $K$ values for each class (specifically, with a larger value of $K$ for the real class).

### 4.1.4 Separability and Alignment in Face Anti-Spoofing (SA-FAS)

SA-FAS (Sun et al., 2023) is a new approach to DG that embraces domain-variant characteristics instead of removing their influence over the feature extraction process. The intuition behind this approach is that removing domain-variant information is not a trivial task (see Figure 4.6), and the authors show that many state-of-the-art methods do not fully accomplish it. What is done in this work is to instead encourage separability and alignment: samples from different domains and classes should have well-separable features, and the transition from live to spoof in the feature space is the same (i.e., aligned) regardless of domain. Modelling the problem as such allows for a domain-invariant decision threshold.

Specifically, due to the limited size of training data, current models cannot achieve a perfectly domain-invariant feature space. Mixing domains in the feature space makes it ambiguous due to domain-specific signals that are carried from each domain, and this can lead the live/spoof classifier to learn unfaithful clues for the task. We can more deeply understand the SAFAS solutions from two different perspectives, namely how it achieves separability and how it achieves alignment.

To achieve separability, the authors propose a Supervised Contrastive Learning scheme, which they name SupCon. The intent in this decision is to learn representations that force samples from the same domain and class (live/spoof) to form a compact cluster. Supervision is adopted here because in this modelling the labels (i.e., class and domain) are known beforehand.

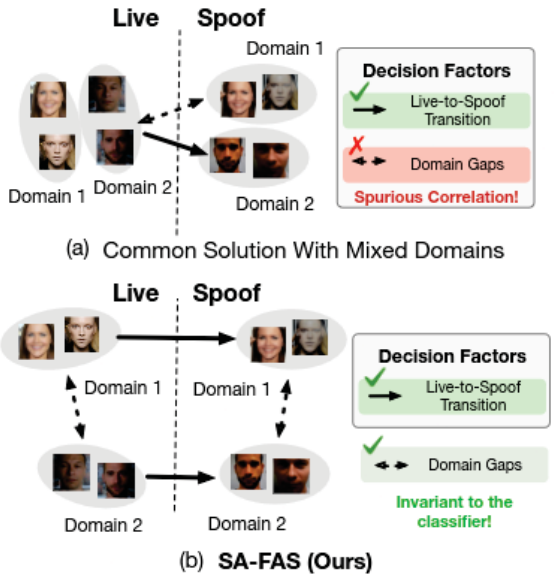



Figura 4.6: Illustration of pitfalls with traditional DG-FAS approaches which obtain representations that carry spurious correlation intrinsically, in comparison with the Separability- and Alignment- oriented SAFAS solution. Source: Sun et al. (2023).

To achieve alignment, the authors model an Invariant Risk Minimization (IRM) problem on regularizing the live-to-spoof transition to enforce it remains invariant to domain variations. The Projected Gradient (PG) optimization strategy is used to solve this optimization objective by taking into consideration its constraints, i.e., keeping the iterative solutions inside the viable set. With PG-IRM, the method optimizes multiple hyperplanes, namely one for each train domain. Finally, with this process, the feature space can be divided between two half-spaces, one for each sample class. The effect of this is that, while domain variations might affect a sample's embedding, they should not affect which half-plane it will belong to. Figure 4.7 illustrates a comparison between this approach and traditional Empirical Risk Minimization.

This method has shown great performance in Domain Generalization tasks, but it has some important limitations that should be pointed out. The first is that it relies on the presence of samples of both classes in each domain, and the second is that PG-IRM will add one hyperplane per domain, which means there is a significant growth in computational cost as we increase the number of domains.

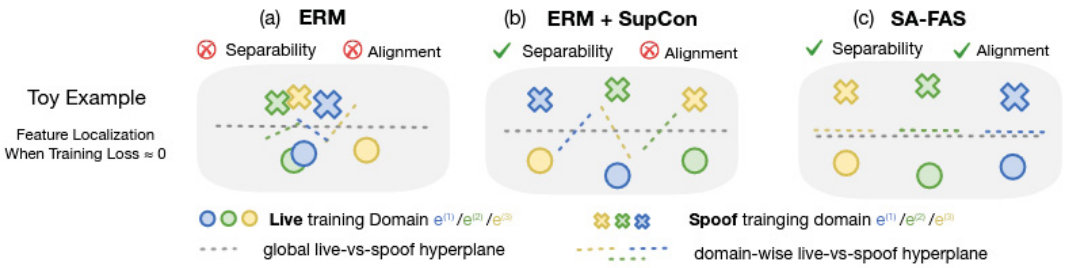



Figura 4.7: Illustrated comparison between Empirical Risk Minimization (e.g., Stochastic Gradient Descent) and Invariant Risk Minimization (IRM). Source: Sun et al. (2023).

### 4.1.5 Dual-Cross Central Difference Convolutional Network (DC-CDN)

The CDCN (Yu et al., 2020a) network was proposed in 2020 and reached state-of-the-art performance in many relevant benchmarks. It consisted of a DepthNet (Liu et al., 2018b) with a simple twist: instead of standard convolutions, it used Central Difference Convolutions. These are convolutions that were shown to have consistent features across shifted domains (see Figure 4.8), which was not the case for standard convolutions, and it could provide great benefits to CNNs in the context of FAS.

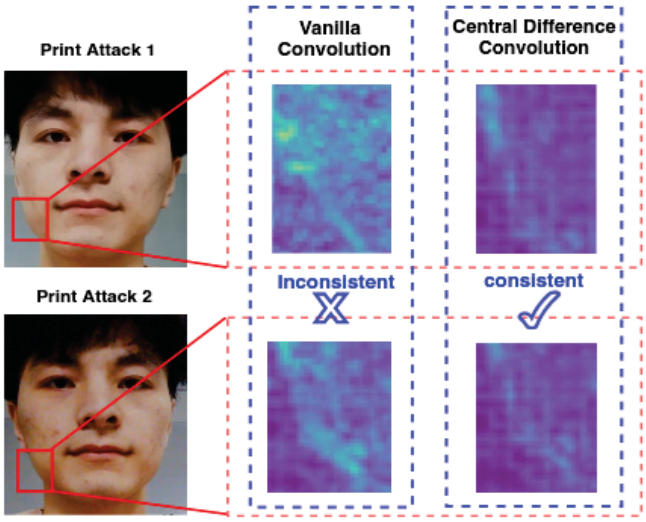



Figura 4.8: Comparison of the feature response between *vanilla* (i.e., traditional) convolutions and the Central Difference Convolution (CDC) for spoofs in shifted illumination and input camera scenarios. Source: Yu et al. (2020a).

These convolutions were fundamentally only slightly different than traditional convolutions, and consisted of the $3 \times 3$ operation with an additional subtraction of the center value afterwards (see Figure 4.9). Additionally, Yu et al. (2020a) use Neural Architecture Search algorithms to further optimize CDCN into CDCN++.

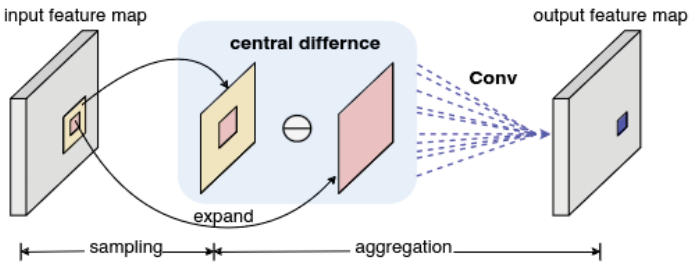



Figura 4.9: Central Difference Convolution (CDC). Source: Yu et al. (2020a).

The DC-CDN architecture builds on CDCN with a very simple twist on CDCs: instead of using the whole $3 \times 3$ grid for the convolution, it was proposed (Yu et al., 2021) to exclude either the horizontal and vertical or diagonal elements, as shown in Figure 4.10. This is done by assuming that the local neighbor region might contain redundancies which lead to overfitting from difficult optimization, and that making this neighbor region sparse (but still center-oriented) could alleviate this issue.

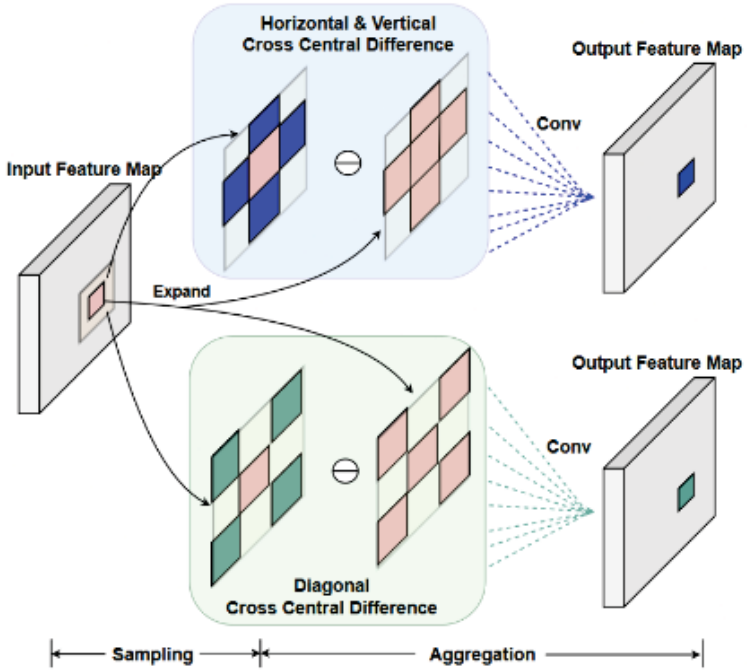



Figura 4.10: Cross Central Difference Convolutions. Source: Yu et al. (2021).

To use both variations of CDCN (with Horizontal/Vertical and Diagonal CDC), DC-CDN consists of two CDCN networks (one with each cross version of CDC) with interacting features at different levels through Cross Feature Interaction Modules (CFIM - see Figure 4.11).

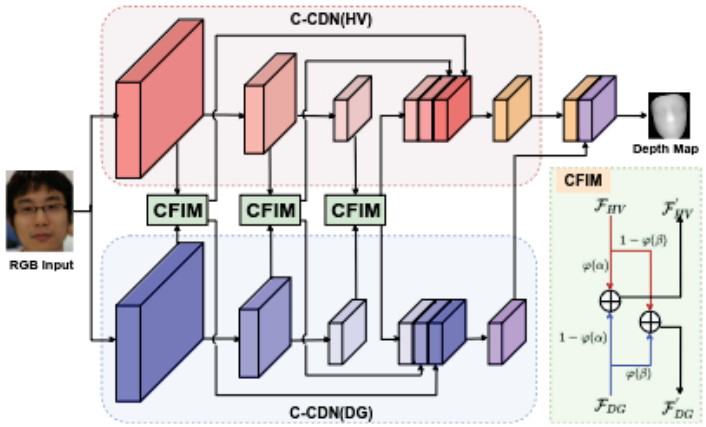



Figura 4.11: DC-CDN network overview. Source: Yu et al. (2021).

### *4.1.5.1 Patch Exchange*

Yu et al. (2021) introduce the Patch Exchange method for data augmentation to enforce locality when training the DC-CDN model (see Figure 4.12). Patch Exchange consists of swapping random rectangular regions between samples (which, in the case of DC-CDN, means swapping the same regions both in face images and their corresponding depth maps). The intuition for the effectiveness of this approach is that, enforced by pixel-wise supervision, the model must learn to determine whether each image region is live or not instead of aggregating global information to produce a depth map as it would a binary label.

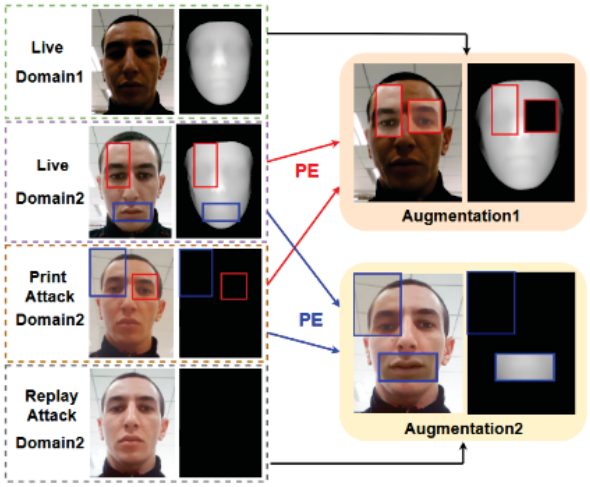



Figura 4.12: Illustration of Patch Exchange. Random rectangular regions are exchanged between a pair of samples, both in the face images and in the corresponding pseudo depth maps. This creates an image with both live and non-live regions. Source: Yu et al. (2021).

## 4.2 PROPOSED AUGMENTATION STRATEGY: LANDMARK EXCHANGE

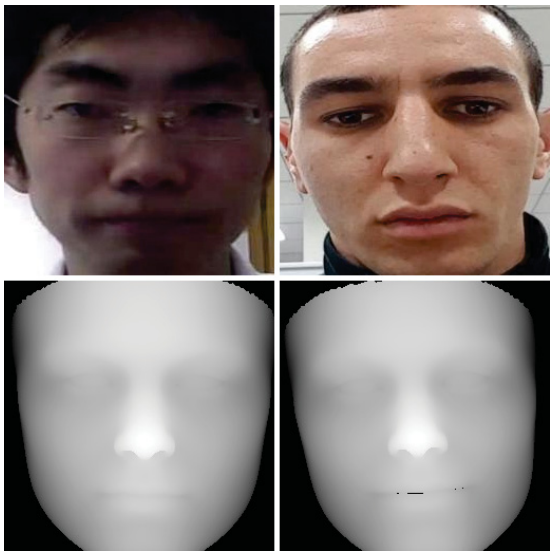
(a) Original samples. The left face is an attack from MSU-MFSD (Wen et al., 2015) and the right is a bona fide sample from Replay-Attack (Chingovska et al., 2012). The spoof has an empty depth map.

(b) Augmented samples generated from the pair by exchanging the left eyebrow polygon and their respective depth maps. Note that the exchange also happens between depth maps, in the same region.

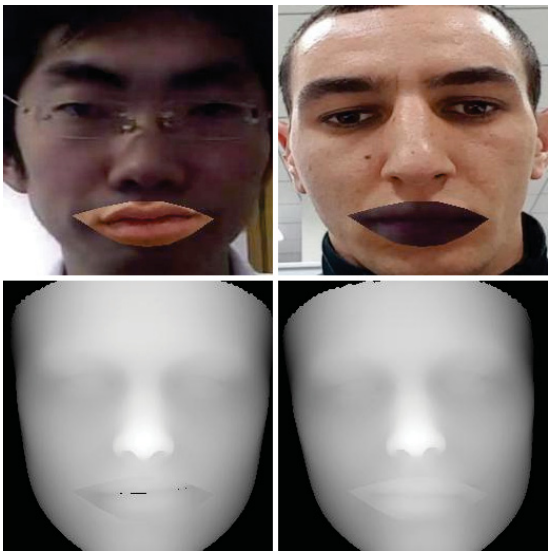
Figura 4.13: Example of landmark exchange between a spoof and a bonafide from MSU-MFSD and Replay-Attack.

We propose to expand on top of Patch Exchange (Yu et al., 2021) (see Figure 4.12) with Landmark Exchange (LMKE), a similar augmentation strategy that exchanges polygons corresponding to random face landmarks (nose, eyes, mouth, eyebrows) instead of random rectangles. The characteristic of exchanging the same patches between the face images and their corresponding depth maps, present in Patch Exchange, is kept for Landmark Exchange.

Patch Exchange enforces model locality by introducing the possibility of different depth properties in different regions of the image. A face might be mostly real but have a small rectangle with spoof characteristics, and so it is not enough for the model to learn to make a



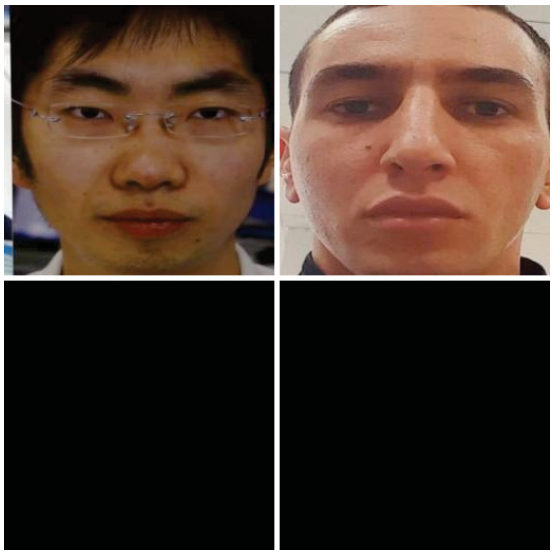
(a) Original samples. The left face is from CASIA-FASD (Zhang et al., 2012) and the right from Oulu-NPU (Boulkenafet et al., 2017). As these are live samples, their depth maps are non-empty.



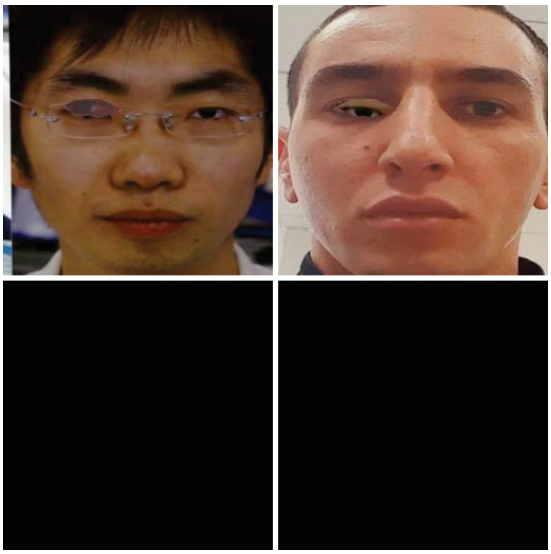
(b) Augmented samples generated from the pair by exchanging the mouth polygon and their respective depth maps. Note that the exchange also happens between depth maps, in the same region.

Figura 4.14: Example of landmark exchange between two real samples from CASIA-FASD and Oulu-NPU.

general judgment and render a "generic" depth map accordingly: one that fits the face region in the case of bona fide samples or an empty one in the case of spoofing attacks.



(a) Original samples. The left and right faces are attacks from CASIA-FASD and Oulu-NPU (Zhang et al., 2012; Boulkenafet et al., 2017), respectively. As they are spoofs, both have empty depth maps.



(b) Augmented samples generated from the pair by exchanging the right eye polygon and their respective depth maps. Note that although the exchange also happens between depth maps, in this case there is no difference since they are both empty.

Figura 4.15: Example of landmark exchange between two spoof samples from CASIA-FASD and Oulu-NPU.

Figures 4.14, 4.13 and 4.15 show examples of patch exchange involving samples from four datasets, CASIA-FASD, MSU-MFSD, Oulu-NPU and Replay-Attack (Zhang et al., 2012; Wen et al., 2015; Boulkenafet et al., 2017; Chingovska et al., 2012).

For obtaining landmarks, we use the 3DDFA-V2 (Guo et al., 2020) network, the same used for extracting pseudo depth maps for bona fide samples. We use most of the landmarks this model can identify, discarding only the jaw points since that would be much larger than other landmark polygons.

## 4.3 MOTIVATION

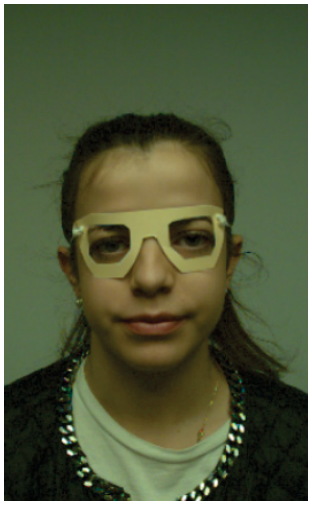
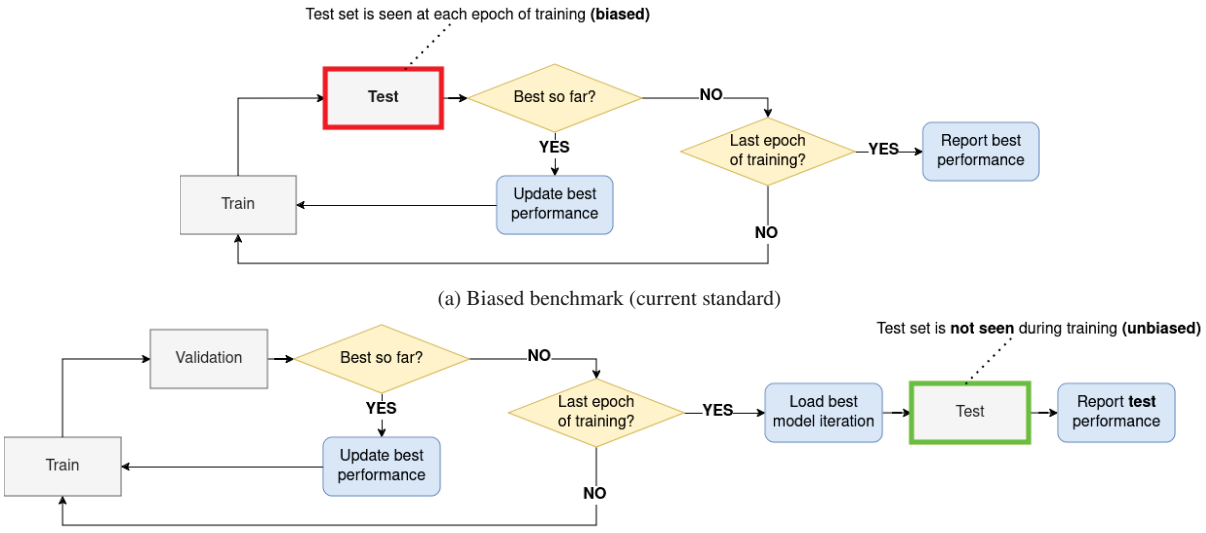


Figura 4.16: An attack that has distinctive characteristics around the eye region. Source: Mostaani et al. (2020).

The idea behind Landmark Exchange is to create the same effect, but instead of using random rectangular regions of the image, we aim to use regions of interest: face landmarks. The intuition behind this approach is that in many attacks, these landmarks will provide discriminative information. Figure 4.16 illustrates this point: in the case of face mask attacks, the eyes or mouth of the attacker are often made visible.

## 4.4 PROPOSED DG-FAS BENCHMARKS




(a) Biased benchmark (current standard)

(b) Proposed unbiased benchmark

Figura 4.17: Biased benchmark (top) vs unbiased (bottom). In the biased benchmark, there is access to the test set in every epoch of training, which leads to biased decision-making (both in the research process and model choice for final evaluation). In our proposed unbiased benchmark, the test set is only used after training is finished, so there is no bias in decision-making.

As briefly discussed in Chapter 2, the current standard benchmark for the evaluation of state-of-the-art methods in DG-FAS consists of evaluating the model on the test set at each

training epoch and then reporting the best results achieved over all epochs. This is a biased approach because access to the test set during training allows for a well-informed model choice, which is not possible in practical applications. Because of this, and since the main motivation of current research in FAS is to apply it to the real world, current evaluation benchmarks are not useful for developing DG-FAS models.

We propose improving the current benchmark to make it unbiased, changing it to make it meet two key requirements: (i) hiding the test set during training, and (ii) allowing for the evaluation of the model's classification and generalization capabilities.

Since the biased benchmark we want to improve consists of four datasets (three for training and one for testing), we propose introducing a fifth dataset as either validation or test. This way, during training the model's generalization and classification capabilities are still evaluated, but the test set does not influence the choice of best-performing model iteration from the training process. When the final model is evaluated on the test set, after training, we have a better representation of its real-world performance. By making the benchmark unbiased, we introduce significant changes to the effects training and testing have on current DG-FAS model research. At the same time, for researchers and practitioners, these changes in practice require simple implementation due to their similarity to already well-established benchmarking procedures. Indeed, for the state-of-the-art models we use in our own experiments, there were no changes to the train/test functions themselves. Figure 4.17 illustrates both biased and our proposed unbiased benchmarks.

# 5 EXPERIMENTS

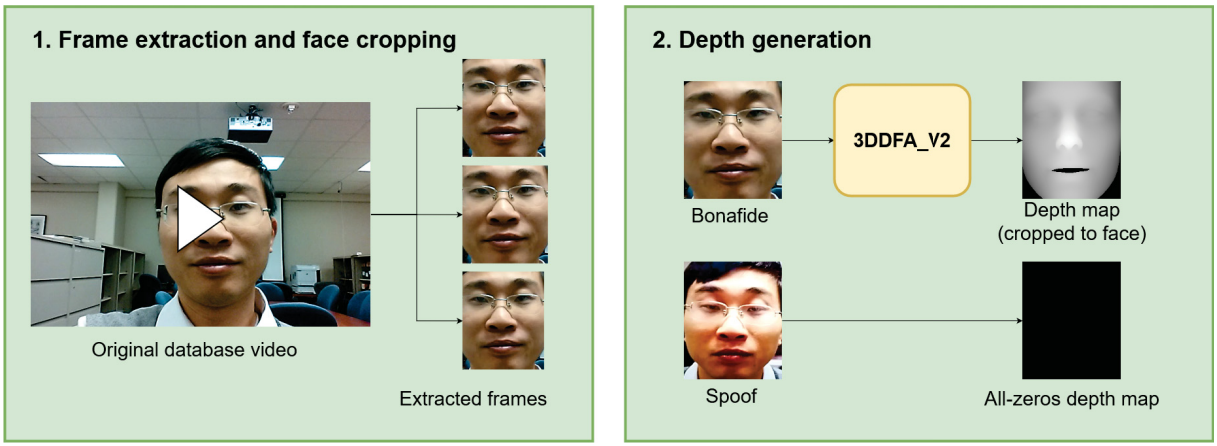



Figura 5.1: Processes for video frame extraction (1) and face depth generation (2). Samples belong to the MSU-MFSD (Wen et al., 2015) dataset. Landmark coordinates obtained to generate the depth maps are saved for later augmentation with LMKE.

This chapter is split into two major sections: Section 5.1 describes the methodology applied for evaluating the proposed architecture, while Section 5.2 presents the obtained results.

## 5.1 METHODOLOGY

In this section, we describe the methodology for treating the data and executing experiments, all of which were executed in one of three different machines with the following available GPUs: NVIDIA GeForce RTX 3060, NVIDIA GeForce RTX 3090, NVIDIA TITAN V (two), NVIDIA GeForce RTX 2080 SUPER, NVIDIA TITAN Xp, and Quadro RTX 8000. Different module versions were used in accordance with each model's requirements, which can be verified in their respective open-source repositories.

### 5.1.1 Datasets

All used datasets were obtained in accordance with the authors' terms of usage and we received official authorization to perform scientific experiments with the provided data. In the case of Oulu-NPU (Boulkenafet et al., 2017), MSU-MFSD (Wen et al., 2015), Replay-Attack (Chingovska et al., 2012), Rose-Youtu (Li et al., 2018), SiW (Liu et al., 2018a) and CASIA-FASD (Zhang et al., 2012), the data is in video format, while the baseline methods (Wang et al., 2022; Zhou et al., 2023; Sun et al., 2023; Yu et al., 2021; Le e Woo, 2024; He et al., 2015) are built to receive images as input. Table 5.1 presents biased benchmarks involving Oulu-NPU, MSU-MFSD, Replay-Attack and CASIA-FASD.

The way this is approached in practice is to sample a frame from each video during each epoch of training, with the frames available for sampling being extracted beforehand for better performance. It is common to extract 8 frames from each video and randomly sample between those 8 frames during training (Wang et al., 2022; Zhou et al., 2023; Yu et al., 2021), but it is also possible to extract more (Sun et al., 2023; Le e Woo, 2024). The extracted frames are treated for face cropping and depth map generation as illustrated in Figure 5.1.

| Name | Type | Train datasets | Test Dataset |
|---|---|---|---|
| ICM to O | Standard | Replay-Attack, CASIA-FASD, MSU-MFSD | Oulu-NPU |
| OCI to M | Standard | Oulu-NPU, Replay-Attack, CASIA-FASD | MSU-MFSD |
| OMI to C | Standard | Oulu-NPU, MSU-MFSD, Replay-Attack | CASIA-FASD |
| OCM to I | Standard | Oulu-NPU, CASIA-FASD, MSU-MFSD | Replay-Attack |
| MI to C | Limited source domain | MSU-MFSD, Replay-Attack | CASIA-FASD |
| MI to O | Limited source domain | MSU-MFSD, Replay-Attack | Oulu-NPU |

Tabela 5.1: Description of biased DG benchmarks used in this work. The first letters are the initials of each dataset in the train set, and the last letter is the initial of the test dataset. *I* corresponds to Replay-Attack (Chingovska et al., 2012). This is a replica of Table 3.2

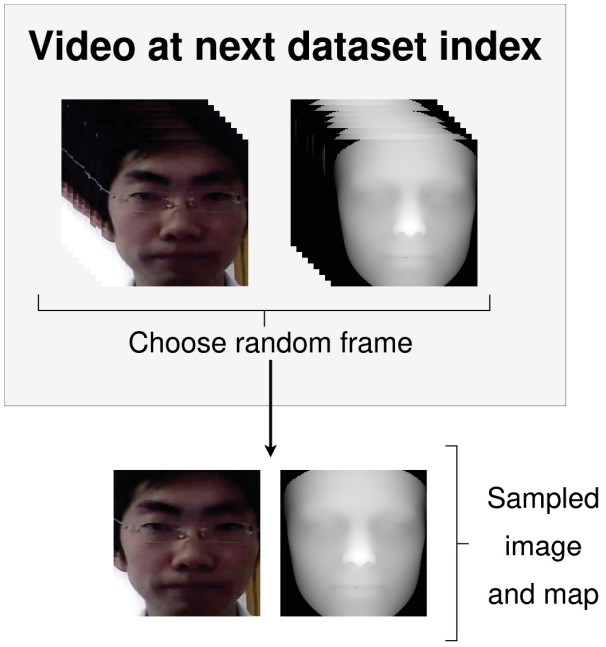



Figura 5.2: Frame sampling illustration. The example image is from the CASIA-FASD (Zhang et al., 2012) dataset.

To generate pseudo depth maps and crop the images to the face region, we use the 3DDFA_V2 (Guo et al., 2020) method, which utilizes a lightweight backbone with a meta-joint optimization strategy to regress a small set of 3DMM (3D Morphable Model, a face representation) parameters and a landmark-regularization regression. Figure 5.2 shows the workflow for extracting frames with depth map and bounding box generation using 3DDFA_V2.

### 5.1.2 Model Training

For LMKE-related benchmarking, we follow the DG evaluation present in recent works (Wang et al., 2022; Zhou et al., 2023; Sun et al., 2023; Le e Woo, 2024) to evaluate model performance in our experiments: after every epoch of training, the model is evaluated on the test set, which consists of data from another dataset, and its performance is recorded. The reported model performance is the best across all epochs. This is different from many other tasks in deep learning, where a validation set is used to determine the model version used in test to report performance. We follow this and use the entire dataset instead of a split (for example, using the entire CASIA-FASD (Zhang et al., 2012) dataset for training instead of its train split) since in DG the train and test data are from different sources. The only exception to this in LMKE

experiments is in our Oulu-NPU-only (Boulkenafet et al., 2017) experiments, which fit into the intra-dataset category.

For experiments related to our unbiased benchmarking proposal, the benchmarks designed in this work are used. They consist of a similar benchmark, only with a validation set for model evaluation at each epoch instead of the test set. The pool of datasets distributed among train, validation and test across different benchmarks consists of the four datasets used in other DG-FAS benchmarks (CASIA-FASD, Replay-Attack, Oulu-NPU and MSU-MFSD) and two additional datasets, namely Rose-Youtu and SiW. In particular, we always use three datasets for training, one for validation and one for testing, which means we always use a total of five datasets in a single benchmark. We highlight that even though our proposed benchmarks are fundamentally different to current state-of-the-art biased benchmarks in DG-FAS, and solve a very important issue, they are not complicated hacks of training procedures - we introduce an elegant design choice that makes for better model training. Figure 4.17 illustrates both the biased DG-FAS benchmarks used as baselines and for LMKE experiments and the proposed unbiased DG-FAS benchmarks in this work.

For each used method, we use the code as it is made available by its authors (Yu et al., 2021; Zhou et al., 2023; Sun et al., 2023; Wang et al., 2022; Le e Woo, 2024). We also have reimplemented DC-CDN and SSAN (Yu et al., 2021; Wang et al., 2022) based on the code made available by the authors, but final results shown here are obtained with the original code. In the case of SA-FAS, IADG and GAC-FAS, (Le e Woo, 2024; Sun et al., 2023; Zhou et al., 2023), we change from a fixed seed for randomized algorithms to a system epoch-based 32-bit integer seed. For experiments with unbiased DG benchmarks we also use ResNet18 and ResNet50 (He et al., 2015), both with implementations from the Torch Image Models library (Wightman, 2019) and pre-trained ImageNet weights.

While comparative experiments in Subsection 5.2.1 may differ from results reported by the paper authors, we emphasize that this work describes in detail the data preparation process and reported results are obtained with code made available by the paper authors (Yu et al., 2021; Wang et al., 2022; Sun et al., 2023; Zhou et al., 2023; Le e Woo, 2024), while most works in the literature (see Chapter 3) do not even mention how the data is treated in their own experiments, which often makes for poor reproducibility of methods (even more so considering that the models here discussed are trained on images extracted from video datasets).

### 5.1.3 Experiments for Exchange Augmentations

Some of the benchmark and model choices in this work involve long experiment durations, and therefore we prioritize experiment execution for qualifying purposes. As a general guideline, we first compare the models in some particular benchmarks, both among themselves and with the results reported in the original papers, and then choose a few models from there to experiment with Patch Exchange (PE) and Landmark Exchange (LMKE) in all benchmarks, which allows for an overview of how these augmentations affect model performance. For specific questions we utilize specific benchmarks and models.

The baseline comparison benchmarks are Oulu-NPU benchmark 1, MI to C (train on MSU-MFSD and Replay-Attack and test on CASIA-FASD) and MIC to O (train on MSU-MFSD, Replay-Attack and CASIA-FASD and test on Oulu-NPU). The Oulu-NPU benchmark is chosen for comparison because it is an intra-dataset task that DC-CDN focuses on (Yu et al., 2021). Although it is out of scope of the other (DG) models, we compare them in that benchmark as well, for intersectionality with the aforementioned model. Among all four Oulu-NPU benchmarks, the first shares very similar characteristics with the general task of DG, namely testing on previously unseen environmental conditions (namely illumination and background scene). MI to C is

chosen as a limited-source DG benchmark, which we hyphotesize can take great advantage of augmentation strategies. MIC to O is chosen as a normal DG benchmark to represent the class of 3-to-1 tasks.

The chosen models for comparison in all benchmarks with exchanges are picked based on their relative performance with respect to other models in all three benchmarks mentioned above.

### 5.1.4 Experiments for Unbiased Benchmarks

We evaluate the proposed benchmark in comparison with four already well-established DG-FAS benchmarks that consist of training on three datasets and testing on a fourth, where the datasets are CASIA-FASD (Zhang et al., 2012), Replay-Attack (Chingovska et al., 2012), MSU-MFSD (Wen et al., 2015) and Oulu-NPU (Boulkenafet et al., 2017). Aditionally we use the WFAS (Wang et al., 2023) dataset for auxiliary experiments on another unbiased benchmark. WFAS differs from other FAS datasets in both number of attacks and volume of data.

In considering options for the fifth dataset, we must consider in which aspects they are similar or different from the four currently used datasets. Particular aspects of interest are the data volume, types of attacks, number of subjects, and variability of samples. We consider two datasets as options for the fifth dataset due to their similarities and differences to the four currently used datasets: SiW (Liu et al., 2018a) and Rose-Youtu (Li et al., 2018). Both involve print and replay attacks, which is the case in the other four datasets used. Rose-Youtu also includes paper mask attack samples, which we discard to keep the task intra-type (no new attacks on the test set). Besides the attack type consideration, we choose these two datasets specifically because they are both reasonably sized (number of samples comparable to that of Oulu-NPU), recent (both more recent than the other four), and varied (both datasets feature variations in illumination, environments, attacks, and subjects; SiW features variations in pose and expression as well). From these characteristics, we hypothesize that they can be at least as useful for DG evaluation as currently used datasets are. Table 5.2 compares the characteristics of these datasets.

We start with baseline experiments on the biased benchmarks that have already been used in previous works. From those experiments, we can establish a reference point for model performance and also compare our own local results with those reported by the authors (in the case of SA-FAS and GAC-FAS). Afterward, we execute three categories of experiments, which we call biased, unbiased with added validation, and unbiased with added test. They corre spond, respectively, to: (1) biased experiments on the additional datasets, (2) unbiased experiments where we add a new dataset as validation in a standard benchmark, and (3) unbiased experiments where we replace the test set with a fifth dataset in a standard benchmark (keeping its previous test set as validation). From these results, we expect to validate our hypothesis that model performance on biased benchmarks (i.e., performance on the test set when there is access to the same test set during training) is not representative of the model's generalization capability, and from the different options of designed benchmarks we draw suggestions for how evaluation should instead be carried out in future work.

We report for each experiment both the test set Half-Total Error Rate and the validation set Equal Error Rate percentages. Additionally, we execute similar experiments when considering WFAS (Wang et al., 2023) as the test set. WFAS is a more recent dataset created from web scraping techniques with the proposal of being larger and more varied than previous FAS datasets and focuses on in-the-wild classification. Unbiased experiments with WFAS differ from those previously described because they consist of a cross-type cross-domain task (a union of the Generalized DL leaf nodes in Figure 2.2), i.e., the WFAS dataset has attack types other than those present in the training datasets. WFAS also has a fundamental difference in size, being huge in

| Dataset | Year | Number of subjects | Live samples | Spoof samples | Attack types |
|---|---|---|---|---|---|
| **CASIA** | 2012 | 50 | 150 | 450 | Print,Replay |
| **Replay** | 2012 | 50 | 200 | 1000 | Print,Replay |
| **MSU** | 2014 | 35 | 70 | 210 | Print,Replay |
| **Oulu** | 2017 | 55 | 720 | 2880 | Print,Replay |
| **SiW** | 2018 | 165 | 1320 | 3300 | Print,Replay |
| **Rose** | 2018 | 20 | 897 | 1601 | Print,Replay,Mask |

Tabela 5.2: Comparison of FAS datasets CASIA-FASD, Replay-Attack, MSU-MFSD, Oulu-NPU, SiW, and Rose-Youtu. Columns correspond, in order, to dataset name, year of publication, number of subjects, number of live samples, number of spoof samples, and attack types (the last being a subset of (P)rint, (R)eplay, and (M)ask attacks). All datasets are video datasets.

comparison to all other datasets (see Table 5.2). The rationale for considering this scenario is to evaluate how well-adapted these models are to a slightly different task. In particular, WFAS consists of 1.383.246 image samples of 469.920 subjects in unconstrained settings, with 17 different types of presentation attacks. Since WFAS includes a range of other attack types and consists of an unconstrained scenario, we consider experiments involving it to be fundamentally different from other experiments in this work. With our unbiased benchmark formulation, we evaluate SA-FAS and GAC-FAS on the WFAS test set after training on standard bench marks (using the test set as validation) and, in the case of GAC-FAS, after training on the WFAS training set. The latter is reserved to GAC-FAS for performance reasons - for comparison, WFAS is almost 300 times larger than SiW, the largest dataset referenced in Table 5.2.

With the WFAS experiments, we aim to verify how applicable these DG-FAS models are to an in-the-wild scenario that compares in many aspects to real-world applications of (even constrained) face recognition. If these models do not indeed perform well in this case, it would reinforce our general hypothesis that current evaluation benchmarks are limited in representing real-world model performance. For experiments with standard DG-FAS benchmarks and our proposed unbiased variations involving SiW and Rose-Youtu, we report the test HTER% and validation EER% values. In experiments involving the WFAS dataset as test set, since it consists of a closed-set evaluation, the APCER%, BPCER% and ACER% results provided by the WFAS ongoing contest (Wang et al., 2023) are shared.

## 5.2 RESULTS

We now go over executed experiments and analyze their results, first for the LMKE proposal and then for the unbiased DG-FAS benchmarks proposal. We don't experiment with unbiased DG-FAS benchmarks and LMKE concurrently because of the unpromising nature we find for LMKE in cross-domain scenarios.

### 5.2.1 Reproduction

We now show some reproduction experiments, i.e., compare results reported by the authors with the ones obtained in our own experiments. All the steps to utilizing the provided code are described in Section 5.1. Results for each method can be seen in Table 5.4. We also compare the performance of both DC-CDN and CDCN++ in our experiments with results reported by authors (Yu et al., 2020a, 2021), in all four Oulu-NPU benchmarks in Table 5.3.

| Benchmark | DC-CDN (paper) | DC-CDN (ours) | CDCN++ (paper) | CDCN++ (ours) |
|---|---|---|---|---|
| 1 | 0.4 | 1.67 | 0.2 | 1.9 |
| 2 | 1.3 | 3.82 | 1.3 | 3.1 |
| 3 | 1.9 ± 1.1 | 1.8 ± 1.3 | 1.8 ± 0.7 | 1.8 ± 1.3 |
| 4 | 4.0 ± 3.1 | 4.2 ± 4.8 | 5.0 ± 2.9 | 3.3 ± 3.4 |

Tabela 5.3: DC-CDN and CDCN++ performance comparison against each other in our own experiments and results reported by authors (Yu et al., 2020a, 2021) on all Oulu-NPU benchmarks. Given results are the ACER% values. In the case of Benchmarks 3 and 4 both the mean and standard deviation are presented.

| Metric | DC-CDN | DC-CDN† | SSAN | SSAN† | SA-FAS | SA-FAS† | IADG | IADG† |
|---|---|---|---|---|---|---|---|---|
| HTER% ↓ at MI to C | 32.80 | - | 28.20 | 30.00 | 24.16 | - | 31.50 | 24.07 |
| ACER% ↓ at O1 | 1.46 | 0.40 | 3.85 | - | 9.27 | - | 5.65 | - |
| HTER% ↓ at MIC to O | 25.81 | - | 16.67 | 19.51 | 12.95 | 10.00 | 17.74 | 8.86 |

Tabela 5.4: Comparison of model performance between reported values in their respective papers and our own local experiments with the code provided by the authors. HTER% values for MRC (train on MSU-MFSD (Wen et al., 2015) and Replay-Attack (Chingovska et al., 2012), test on CASIA-FASD (Zhang et al., 2012)) and MRCO (train on MSU-MFSD, Replay-Attack and CASIA-FASD (Wen et al., 2015; Chingovska et al., 2012; Zhang et al., 2012) and test on Oulu-NPU (Boulkenafet et al., 2017)) benchmarks and ACER% value for Oulu-NPU Benchmark 1 (O1) (Boulkenafet et al., 2017) for each model. A value of – indicates the paper does not report the model performance for a particular scenario. Columns with the † mark correspond to results reported by the authors (Yu et al., 2021; Wang et al., 2022; Sun et al., 2023; Zhou et al., 2023) while the rest correspond to our own experiments.

We highlight that these reproduction experiments come from very carefully treated data and mostly unaltered original code shared by the authors of each cited paper. More details about the reproduction experiments are in Section 5.1.

From these results, we choose the model SA-FAS (Sun et al., 2023) for experimenting with Patch Exchange and Landmark Exchange due to its generally better-than-others performance in Table 5.4 above when taking into consideration all three benchmarks. Another reason is that this is a very recently proposed approach to DG-FAS, and is therefore the most updated in terms of research trends, leading to better similarities to other models that might benefit or not from the proposed augmentation method. Finally, it allows for investigating the effect of these augmentation when a model performs well as well as poorly (which are the cases of the DG benchmarks and the first Oulu-NPU benchmark, respectively).

As explained in Chapter 2, one of the aspects of training that allow models to leverage the Patch Exchange augmentation is pixelwise supervision over pseudo depth maps. SA-FAS does not take advantage of this, and so we also experiment with DC-CDN and CDCN++. One limiting factor for experimenting with these networks is the computational cost of model training. We try adjustments to technicalities but the model complexity itself is very high due to the nature of these networks and so we are unable to run experiments on all variations of benchmarks (in comparison with SA-FAS, for example). Table 5.5 shows for example how for the computation of the Contrastive Depth Loss switching from creating kernels in CPU (with NumPy, as in the code made available by the authors of DC-CDN and CDCN++) to creating them directly in GPU (with PyTorch) only enables an insignificant, constant reduction in time complexity.

For comparison of biased and unbiased DG-FAS benchmarks, Table 5.6 shows results on the standard biased benchmarks used in previous works. We compare our own local results on SA-FAS and GAC-FAS with results reported in recent state-of-the-art papers, and include the GAC-FAS results obtained when using authors' provided weights. Even though we used the code and datasets as they were publicly provided by the authors, we once again highlight that error rates have differences.

| Number of batches | NumPy Time | PyTorch Time |
|---|---|---|
| $10^5$ | 10.3 | 9.7 |
| $5 \cdot 10^5$ | 46.3 | 43.8 |
| $10^7$ | 899.2 | 855.1 |

Tabela 5.5: Time comparison in seconds for generating the contrastive loss kernel for different numbers of batches in NumPy or directly in PyTorch. As expected, the constant complexity of data migration from CPU to GPU does have a negative impact on execution time, but not a significant one.

| GAC-FAS | ICM→O | | OCM→I | | OCI→M | | OMI→C | |
|---|---|---|---|---|---|---|---|---|
| | HTER%↓ | AUC%↑ | HTER%↓ | AUC%↑ | HTER%↓ | AUC%↑ | HTER%↓ | AUC%↑ |
| **SSAN**† | 13.72 | 93.63 | 8.88 | 96.79 | 6.67 | 98.75 | 10.00 | 96.67 |
| **IADG**† | 8.86 | 97.14 | 10.62 | 94.50 | 5.41 | 98.19 | 8.70 | 96.40 |
| **SA-FAS**† | 10.00 | 96.23 | 6.58 | 97.54 | 5.95 | 96.55 | 8.78 | 95.37 |
| **GAC-FAS**† | 8.60 | 97.16 | 4.29 | 98.87 | 5.00 | 97.56 | 8.20 | 95.16 |
| **GAC-FAS**$^w$ | 9.72 | 96.87 | 5.13 | 99.02 | 5.00 | 97.54 | 10.00 | 94.76 |
| **GAC-FAS** | 10.52 | 96.27 | 5.00 | 98.36 | 7.92 | 96.31 | 12.22 | 93.09 |
| **SA-FAS** | 12.03 | 94.78 | 6.55 | 97.66 | 8.57 | 96.96 | 13.33 | 92.29 |

Tabela 5.6: Reproduction results: test HTER% and AUC% results on standard biased DG-FAS benchmarks reported in related works (top, rows marked with †) and obtained in our own experiments (bottom). We include the GAC-FAS evaluation with provided weights (*GAC-FAS*$^w$). Benchmark names correspond to the training datasets (first three letters) and the test dataset (last letter), where I = Replay-Attack, C = CASIA-FASD, M = MSU-MFSD, and O = Oulu-NPU.

### 5.2.2 Exchange Augmentations

We now explore experiments with SA-FAS, DC-CDN and CDCN++ in some benchmarks for Domain Generalization and Intra-Dataset evaluation, with varying exchange augmentations - namely no augmentations, Patch Exchange and Landmark Exchange. We explore some of the parameters for using these techniques and how they affect model performance in each scenario.

The first experiments are simple usages of these techniques for model training by duplicating the train set size with the addition of augmented samples. Afterwards, we observe how a different **ratio** (the number of added augmented samples in comparison with the original train set size) changes the model's effectiveness for classification, and do the same analysis for labeling augmented samples according to different rules.

In particular, we experiment with two possibilities for labeling augmented samples. The first is to *inherit* the label from the "parent" samples: if an exchange sample $A$ is created from two original samples $B$ and $C$, then the label for $A$ will be *live* only if both $B$ and $C$ are live, and *spoof* if not. The idea behind this strategy is that if at least one of $B$ and $C$ is a spoof, then $A$ will contain some traces of spoofs, and if not, then $A$ will contain no traces of spoofs. The second strategy is to assign the *spoof* label to all exchange-created samples, since they contain image manipulation and do not seem like pictures to the human eye. We refer to this strategy as *aug=spoof*. In our experiments, we use the *inherit* strategy by default, and highlight the switch to the second option when it is used.

We also seek to enhance results with artificial class balancing in Domain Generalization benchmarks and visualize results for some experiments in both Domain Generalization and Intra-Dataset evaluation. For these visualizations, we sort test samples according to the distance $|c - y|$ between the classification score $c$ and the label $y$, and samples with the smallest and greatest

values for each scenario and label are chosen. This visualization is done in two benchmarks, namely OCM to I (DG) and Oulu-NPU benchmark 1 (Intra-Dataset).

#### *5.2.2.1 Experiments on Domain Generalization Benchmarks*

| Benchmark | Authors' | Ours | Ours (PE) | Ours (LMKE) |
|---|---|---|---|---|
| MI to C | - | 24.16 | 24.69 | 33.62 |
| MI to O | - | 21.98 | 24.43 | 22.06 |
| ICM to O | 10.00 | 12.95 | 17.42 | 16.65 |
| OMI to C | 8.78 | 13.98 | 15.49 | 23.87 |
| OCM to I | 6.58 | 6.91 | 16.32 | 9.42 |
| OCI to M | 5.95 | 11.14 | 11.05 | 12.91 |

Tabela 5.7: SA-FAS HTER% (for DG benchmarks) mean values obtained in our experiments, as well as the values reported by the authors (Sun et al., 2023) (first column).

We show results for our experiments on Domain Generalization benchmarks involving CASIA-FASD, MSU-MFSD, Oulu-NPU and Replay Attack. We start with an overview of the SA-FAS HTER% in four 3-to-1 benchmarks and two limited train set benchmarks (MI to C and MI to O). Table 5.7 compares the performance of this model in our experiments with no exchange augmentations and with PE and LMKE, as well as the original results reported by Sun et al. (2023). As can be seen, neither exchange augmentation has a positive impact on SA-FAS performance, with inconsistencies between relative performance effects between PE and LMKE. This can be explained by the absence of pixelwise supervision in this method. Due to this, model training does not leverage modifications that exchange augmentations introduce, and the characteristic visual differences in the face image of augmented samples might hinder convergence.

| Exchange | Ratio | aug=spoof | OCM to I performance | | |
|---|---|---|---|---|---|
| | | | Mean | Std | Min |
| **None** | 0 | No | 6.91 | 1.27 | 4.00 |
| **PE** | 1 | No | 16.32 | 1.92 | 14.50 |
| | 1 | Yes | 9.19 | 1.19 | 7.62 |
| | 2 | Yes | 7.50 | 1.87 | 5.50 |
| | 3 | Yes | 12.63 | 1.21 | 11.00 |
| | 6 | Yes | 14.97 | 1.67 | 12.90 |
| | 6 | No | 11.48 | 1.55 | 9.00 |
| **LMKE** | 1 | No | 9.42 | 1.74 | 7.10 |
| | 1 | Yes | 14.37 | 1.06 | 13.00 |
| | 2 | Yes | 9.03 | 1.63 | 7.05 |

Tabela 5.8: SA-FAS mean, minimum and standard deviation of HTER% obtained values on benchmark OCM to I (train on MSU-MFSD, Oulu-NPU and CASIA-FASD (Wen et al., 2015; Boulkenafet et al., 2017; Zhang et al., 2012) and test on Replay-Attack (Chingovska et al., 2012)) with variations in augmentations. *aug=spoof* indicates that all augmented samples are labeled as spoofs and the *ratio* value indicates how many augmented samples were used in the experiment in comparison with the original training dataset (so a ratio of 2 indicates that there are twice as many augmented samples as there are original samples).

Next, in Table 5.8 we explore, for benchmark OCM to I, the introduction of different ratios and the usage of the *aug=spoof* labeling strategy. Although this strategy is different from

how Patch Exchange was first used (Yu et al., 2021), our results show how it can improve model performance under exchange augmentations. As for ratio changes, we observe that an extensive usage of augmented samples is not as beneficial as smaller ratios, namely those between 1 and 3. However, combining a ratio of 1 with *aug=spoof* worsens results in nearly every other DG benchmark, with the exception being MI to O, as shown in Table 5.9.

Tabela 5.9: SA-FAS performance with single label (all augmented samples are considered spoofs) and ratio 1 in all 6 DG benchmarks, with both PE and LMKE.

| Augmentation | MI to C | MI to O | ICM to O | OMI to C | OCM to I | OCI to M |
|---|---|---|---|---|---|---|
| PE | 26.67 | 24.14 | 16.57 | 14.11 | 13.50 | 10.00 |
| LMKE | 28.00 | 19.80 | 15.56 | 25.33 | 15.40 | 14.29 |

Largely increasing the train set with augmented samples of a single label could cause class imbalance and negatively impact model training, and in Table 5.10 we explore the usage of artificial class balancing. Unfortunately, we find no pattern in behaviour caused by the introduction of this technique.

| Setting | Class balancing | aug=spoof | ICM to O | OMI to C | OCM to I | OCI to M |
|---|---|---|---|---|---|---|
| Paper | No | No | 10.00 | 8.78 | 6.58 | 5.95 |
| Ours | No | No | 12.95 | 13.98 | 6.91 | 11.14 |
| Ours + PE | Yes | Yes | 14.04 | 14.67 | 9.45 | 12.86 |
| | | No | 18.69 | 15.33 | 8.45 | 11.43 |
| Ours + LMKE | | Yes | 15.67 | 22.56 | 12.05 | 15.48 |
| | | No | 12.22 | 18.00 | 13.00 | 15.48 |

Tabela 5.10: SA-FAS performance comparison with the introduction of class balancing together with exchange augmentations, with and without the single label rule for augmented samples, in four 3 to 1 DG benchmarks.

In Figure 5.3 we observe some examples of samples correctly and incorrectly classified by SA-FAS for the OCM to I benchmark, with no exchange augmentations, Patch Exchange and Landmark Exchange. One bona fide sample is the “most difficult” (i.e., with a model-assigned score furthest from its true label) in two of the three rows, two spoof samples from the same subject are present in two different rows (note that they are not the same spoof sample), and one subject is present in different rows - one as a correctly labeled bona fide and another as an incorrectly labeled spoof -, but not a lot of insight can be obtained from this as the test sets in these DG benchmarks do not have a large number of subjects (see Table 3.1).

Table 5.11 in a similar division to that of Figure 5.3 shows the count and percentage of incorrectly classified samples per label in this benchmark. We find that percentages in the same row are very similar, which indicates the model does not become biased towards a particular class.

#### *5.2.2.2 Experiments on Intra-Dataset Benchmarks*

We start with a comparison of SA-FAS performance in Oulu-NPU benchmark 1 with varying augmentations - no exchange, Patch Exchange and Landmark Exchange. Table 5.12 shows how model performance is improved by Patch Exchange and further enhanced by Landmark Exchange. We follow by experimenting with DC-CDN and CDCN++, as the conceptualization of these two models is very close to that of Patch Exchange itself (Yu et al., 2020a, 2021) and they were designed with intra-dataset evaluation in mind. Table 5.13 compares the performance

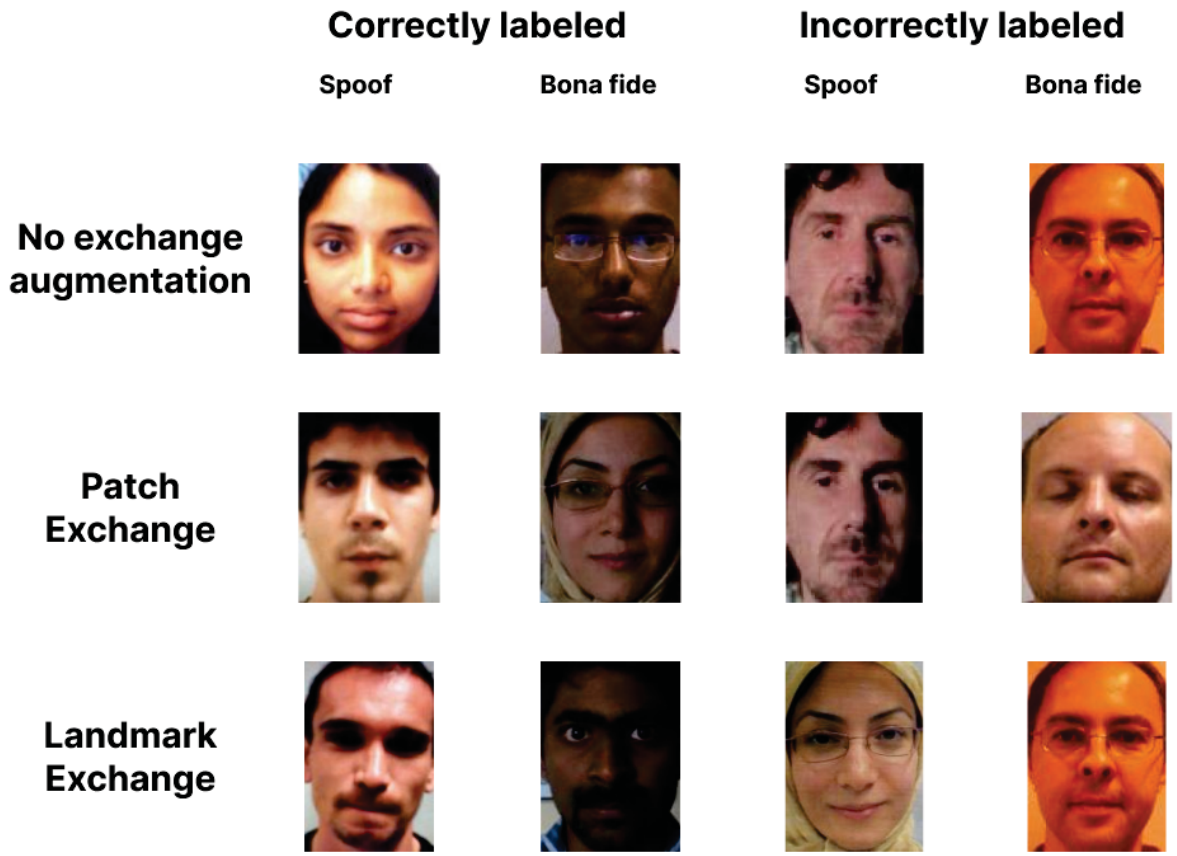



Figura 5.3: SA-FAS worst and best classifications for each label in the test set of OCM to I with no exchange augmentations, PE and LMKE.

| Variant | Spoof | | Bona fide | |
|---|---|---|---|---|
| | Count | Percentage | Count | Percentage |
| No exchanges | 104 | 10.4 | 21 | 10.5 |
| Patch Exchange | 131 | 13.1 | 27 | 13.5 |
| Landmark Exchange | 200 | 20.0 | 41 | 20.5 |
| Total in test set | 1000 | 100.0 | 200 | 100.0 |

Tabela 5.11: Count of incorrectly classified samples by label in OCM to I for SA-FAS with no exchange augmentations, PE and LMKE.

| Benchmark | Authors' | Ours | Ours (PE) | Ours (LMKE) |
|---|---|---|---|---|
| Oulu-1 | - | 9.27 | 9.21 | 8.84 |

Tabela 5.12: SA-FAS ACER% for Oulu-NPU benchmark 1 mean values obtained in our experiments. The authors do not report results in this benchmark (Sun et al., 2023).

of these two networks with each of the exchange augmentations in question in benchmark 1 of Oulu-NPU and we find that, in a similar manner to SA-FAS, CDCN++ benefits from the exchange augmentations.

| Augmentation | DC-CDN | CDCN++ |
|---|---|---|
| None | 1.67 | 1.87 |
| PE | 2.50 | 1.67 |
| LMKE | 2.19 | 1.56 |

Tabela 5.13: DC-CDN and CDCN++ performance on Oulu-NPU benchmark 1, with Patch Exchange, Landmark Exchange and no exchange augmentations. PE and LMKE experiments are carried out with a ratio of 1

We also explore the usage of the *aug=spoof* strategy in the intra-dataset scenario and evaluate how SA-FAS ACER% on Oulu-NPU benchmark 1 changes with the introduction of this technique. Table 5.15 shows results of these experiments, and it can be seen that in this case error rates skyrocket.

| Augmentation | ACER% |
|---|---|
| PE | 7.31 |
| LMKE | 10.17 |

Tabela 5.14: SA-FAS performance on Oulu-NPU benchmark 1, with both Patch Exchange and Landmark Exchange, with an augmentation ratio of 4.

Finally, Figure 5.4 shows examples of correctly and incorrectly classified samples in Oulu-NPU benchmark 1, without exchange augmentations with Patch Exchange and Landmark Exchange, for the SA-FAS model. One subject has “easy” (correctly labeled and score very close to the true label) spoof samples shown in the first and third rows, and another has a bona fide sample that is incorrectly labeled in all three experiments. Other visualized ocurrences are varied and present no subject repetition.

Tabela 5.15: SA-FAS result comparison (ACER%) in Oulu-NPU benchmark 1 with and without augmentations. *aug=spoof* indicates that all augmented samples are treated as spoofs.

| aug=spoof | Exchange | ACER% |
|---|---|---|
| No | None | 9.27 |
| | PE | 9.21 |
| | LMKE | 8.84 |
| Yes | PE | 47.94 |
| | LMKE | 44.88 |

In Table 5.16 we verify how many samples are incorrectly classified per label in the same division of (no exchanges, PE and LMKE) as in Figure 5.4. We can see that there are no substantial variations in neither count nor percentage of incorrectly labeled samples, which indicates that although some samples might be less indicative of their class (such as the repeating bona fide in Figure 5.4), the model itself shows balanced performance across classes in general.

### 5.2.3 Biased and Unbiased Benchmarks

We now share and discuss results and experiments for our studies on bias in DG-FAS benchmarks. We start with biased benchmarks that differ from the current state of the art (see Table 5.6), and then move on to unbiased alternatives.

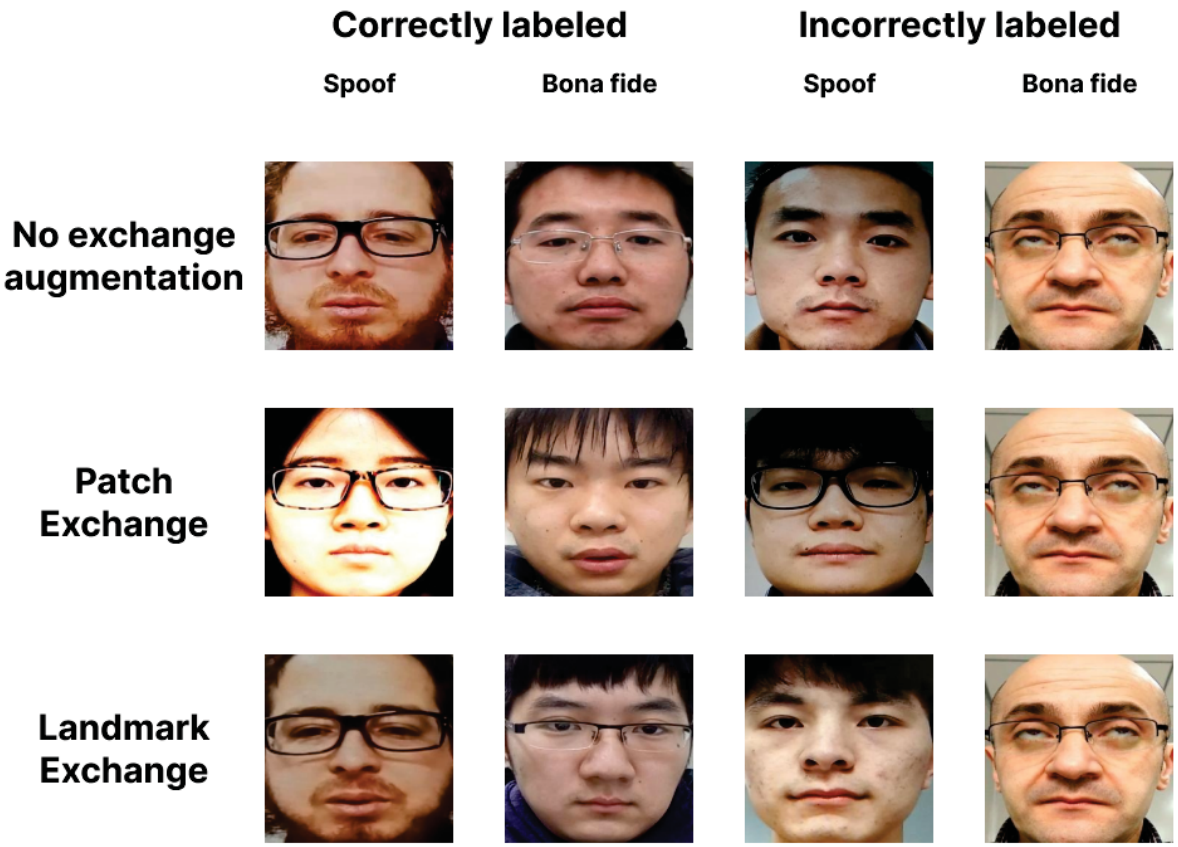



Figura 5.4: SA-FAS worst and best classifications for each label in the test set of Oulu-NPU benchmark 1 with no exchange augmentations, PE and LMKE.

| Variant | Print Attack | | Replay Attack | | Bona Fide | |
|---|---|---|---|---|---|---|
| | Count | Percentage | Count | Percentage | Count | Percentage |
| No exchanges | 16 | 6.6 | 23 | 9.5 | 10 | 8.3 |
| Patch Exchange | 19 | 7.9 | 24 | 10.0 | 11 | 9.1 |
| Landmark Exchange | 20 | 8.3 | 23 | 9.5 | 11 | 9.1 |
| Total in test set | 240 | 100.0 | 240 | 100.0 | 120 | 100 |

Tabela 5.16: Count of incorrectly classified samples by label and attack type in Oulu-NPU benchmark 1 for SA-FAS with no exchange augmentations, PE and LMKE.

#### *5.2.3.1 Biased Benchmarks*

In Table 5.17 we show model performance for biased benchmarks (rows where the validation and test sets are equal). In particular, we experiment with biased benchmarks where the testing set is SiW or Rose-Youtu. We observe that, among biased options, testing on Rose-Youtu is generally more challenging than other datasets. At the same time, SiW is not particularly challenging in comparison to the other options. This holds for more complex DG-FAS models SA-FAS and GAC-FAS as much as it does for the baseline ResNet models, even though ResNet models do not as consistently follow the performance trends of DG-FAS models due to their being less specialized methods. These biased results allow us to make other observations on the basis that, in terms of variability in comparison to other domains, neither SiW nor Rose-Youtu are too simple (to make the task trivial) or too complex (making the task impossible in practice) datasets in comparison to CASIA-FASD, Replay-Attack, MSU-MFSD and Oulu-NPU.

| Train | Val | Test | RN18 | | RN50 | | SA | | GAC | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | V-EER%↓ | T-HTER%↓ | V-EER%↓ | T-HTER%↓ | V-EER%↓ | T-HTER%↓ | V-EER%↓ | T-HTER%↓ |
| ICM | O | | 30.56 | | 33.33 | | 12.13 | | 10.52 | |
| | S | | 21.55 | | 20.57 | | 7.17 | | 7.62 | |
| | R | | 22.50 | | 23.09 | | 20.97 | | 19.29 | |
| | S | O | 21.55 | 32.50 | 20.57 | 32.78 | 7.17 | 13.81 | 7.62 | 7.91 |
| | R | O | 22.50 | 33.33 | 23.09 | 31.74 | 20.97 | 13.64 | 19.29 | 19.94 |
| | O | S | 30.56 | 25.06 | 33.33 | 22.93 | 12.13 | 8.76 | 10.52 | 12.04 |
| | O | R | 30.56 | 25.66 | 33.33 | 24.21 | 12.13 | 20.82 | 10.52 | 19.38 |
| OCM | I | | 31.50 | | 33.75 | | 9.00 | | 5.00 | |
| | S | | 18.67 | | 18.89 | | 5.18 | | 8.15 | |
| | R | | 24.42 | | 16.05 | | 24.10 | | 21.38 | |
| | S | I | 18.67 | 37.50 | 18.89 | 33.75 | 5.18 | 21.50 | 8.15 | 8.55 |
| | R | I | 24.42 | 41.25 | 16.05 | 45.00 | 24.10 | 20.10 | 21.38 | 21.41 |
| | I | S | 31.50 | 21.03 | 33.75 | 26.12 | 9.00 | 6.48 | 5.00 | 11.28 |
| | I | R | 31.50 | 27.10 | 33.75 | 22.06 | 9.00 | 26.86 | 5.00 | 29.86 |
| OMI | C | | 17.96 | | 22.22 | | 14.00 | | 12.22 | |
| | S | | 19.36 | | 19.20 | | 7.93 | | 9.52 | |
| | R | | 19.18 | | 20.29 | | 21.50 | | 19.97 | |
| | S | C | 19.36 | 18.89 | 19.20 | 25.56 | 7.93 | 26.67 | 9.52 | 9.14 |
| | R | C | 19.18 | 22.22 | 20.29 | 23.33 | 21.50 | 40.67 | 19.97 | 20.62 |
| | C | S | 17.96 | 24.07 | 22.22 | 17.78 | 14.00 | 14.09 | 12.22 | 12.95 |
| | C | R | 17.96 | 20.62 | 22.22 | 21.41 | 14.00 | 27.42 | 12.22 | 32.10 |
| OCI | M | | 24.58 | | 32.08 | | 11.43 | | 7.92 | |
| | S | | 22.40 | | 19.65 | | 6.32 | | 7.08 | |
| | R | | 20.18 | | 19.18 | | 19.18 | | 22.98 | |
| | S | M | 22.40 | 32.50 | 19.65 | 25.42 | 6.32 | 11.43 | 7.08 | 7.31 |
| | R | M | 20.18 | 32.50 | 19.18 | 30.00 | 19.18 | 15.48 | 22.98 | 22.42 |
| | M | S | 24.58 | 25.68 | 32.08 | 22.24 | 11.43 | 10.05 | 7.92 | 9.58 |
| | M | R | 24.58 | 22.74 | 32.08 | 21.82 | 11.43 | 25.18 | 7.92 | 32.74 |

Tabela 5.17: Test HTER and Validation EER (T-HTER and V-EER, reported in percentages) for ResNet18, ResNet50, SA-FAS and GAC-FAS on different standard and proposed benchmarks (described by the datasets used for training, validating and testing, where I=Replay-Attack, C=CASIA-FASD, M=MSU-MFSD, O=Oulu-NPU, S=SiW and R=Rose-Youtu). Rows where the validation and test values are the same correspond to biased benchmarks. The first row of each group corresponds to standard benchmarks already used in previous works, and all other rows correspond to some benchmark proposed in this work. We highlight unbiased benchmarks with introduced test sets since they differ in the final evaluation dataset from standard biased benchmarks.

#### 5.2.3.2 *Unbiased Benchmarks*

Also in Table 5.17 we have the model performance for proposed unbiased benchmarks with the addition of SiW or Rose-Youtu as either a validation or a testing dataset (rows where the validation and test sets are different). From both options (adding them as validation or test, i.e., light gray rows), in general, we observe that the proposed unbiased benchmarks yielded higher error rates (HTER), which is expected without the test set bias. In general, removing bias from any of the biased benchmarks by adding some dataset as either validation or test set yields worse model performance, which hints at defective learning. Additionally, as in biased benchmarks, Rose-Youtu shows a higher decay in performance than SiW. Note that, even though adding the fifth dataset as a test set produces a similar behavior as adding it as a validation set, the research community might be more interested in the latter option since that allows for error reporting in the original test sets (Oulu-NPU, Replay-Attack, CASIA- FASD, and MSU-MFSD), which have been of interest to this community for many years (Yu et al., 2022).

Furthermore, notice that the experiments where SiW and Rose-Youtu are added as test sets may be seen as using models trained in standard biased benchmarks and testing them on yet another dataset. This is a general trend. If we look in detail, however, it is possible to notice that in three out of four experiments using SiW as validation when training GAC-FAS, which is the best-performing model used in our experiments, the test set performance is even better than in biased experiments. Specifically, for the benchmarks ICMO, OMIC, and OCIM, using SiW as validation renders GAC-FAS better test performance (i.e., evaluated generalization capability) than in previous standard benchmarks. In the case of ICMO, the unbiased experiment obtained 7.91% HTER which is even better than the state-of-the-art 8.60% HTER reported by the authors (Le e Woo, 2024).

We also note that SiW brings similar performance improvements for ResNet50 in ICMO and OCIM, but we highlight GAC-FAS because it is the current state- of-the-art for many benchmarks.

**Validation EER analysis** As the state-of-the-art benchmarks we use as the basis for our unbiased benchmarks proposal derive the EER threshold from the validation set, the EER we report here is the same as the HTER on the biased benchmarks - based on the training sets and the biased test set. We believe, however, that the EER results may better provide insights into the training procedure, considering our other shared results are focused on the test set performance. In general, we observe that validation performance is better than test performance; this is coherent with machine learning literature (Yu et al., 2022) and with our previous hypothesis because a validation set can be informally described as a test set that is accessible outside the testing phase. In a few experiments involving Rose-Youtu as the fifth dataset, however, the validation performance is consistently worse than the test performance, which is not incoherent, since using different domains for validation and testing exposes us to the possibility that the model cannot bridge the gap between validation and test domains. In particular, we hypothesized that the data domain of the SiW dataset is closer to that of the testing dataset in the benchmark than to that of the Rose-Youtu dataset.

#### 5.2.3.3 *WFAS-based unbiased experiments*

Table 5.18 shows the ACER% performance of SA-FAS and GAC-FAS when trained on different sources and tested on the closed test set of WFAS (Wang et al., 2023)[1]. Note that

[1]Since we are dealing with a closed test set, we only have access to test set performance in three metrics: APCER, BPCER, and ACER.

in WFAS the ACER% is reported, as opposed to other benchmarks, where we instead employ the HTER% metric. The ACER value is often chosen for intra-dataset evaluation due to its worst-case representability: it is very similar to the HTER, but calculated considering each attack type subset of the data in isolation (and the worst performance is the reported one) (Yu et al., 2022). For cross-dataset source benchmarks, both models perform evenly, with SA-FAS showing better results in two benchmarks and GAC-FAS showing better results in the other two. It must be noted, however, that these results are very far from the 2.82% ACER of the winner method for the WFAS challenge (Wang et al., 2023), which was designed specifically for the intra-dataset task of WFAS. This is both due to the different nature of the WFAS task and due to the more challenging characteristic of our proposed unbiased benchmark. Additionally, we show the validation EER% performance for experiments based on cross-dataset benchmarks of Table 5.18 in Table 5.19. In the intra-dataset experiment of Table 5.18, we use the most recent instead of the best model iteration (Le e Woo, 2024) for evaluation on the test set to use the full training set of WFAS.

With these results, we highlight how fewer constraints in the data acquisition process (which is the case of WFAS, a much more varied dataset than all others mentioned in this work) reveal weaknesses in these DG-FAS methods. Even though these models are not specifically designed for all the attack types and the intra-dataset task of WFAS, we hope they are designed with application to real-world systems in mind. And so it is important to, again, consider how effective our evaluation systems are in estimating model performance in production.

| Train | Val | SA | GAC |
|---|---|---|---|
| **ICM** | **O** | 30.14 | 31.99 |
| **OCM** | **I** | 32.54 | 28.62 |
| **OMI** | **C** | 28.67 | 28.76 |
| **OCI** | **M** | 31.22 | 30.76 |
| **WFAS** | **WFAS** | - | 21.51 |

Tabela 5.18: SA-FAS and GAC-FAS performance (ACER%) in the WFAS test set when trained (biased) on given train and validation datasets, where I = Replay-Attack, C = CASIA-FASD, M = MSU-MFSD and O = Oulu-NPU. For time constraint reasons, we only experiment with training on WFAS (last row) with GAC-FAS. Note that the last row corresponds to an **intra-dataset** evaluation, while the others are **cross-dataset**.

| Train | Val | SA | GAC |
|---|---|---|---|
| **ICM** | **O** | 12.13 | 10.52 |
| **OCM** | **I** | 9.00 | 5.00 |
| **OMI** | **C** | 14.00 | 12.22 |
| **OCI** | **M** | 11.43 | 7.92 |

Tabela 5.19: SA-FAS and GAC-FAS validation performance (EER%) for cross-dataset experiments in Table 5.18, where I = Replay-Attack, C = CASIA-FASD, M = MSU-MFSD and O = Oulu-NPU.

## 5.3 CONCLUDING REMARKS

We have presented the detailed methodology used in this work and the results obtained through the executed experiments.

Regarding LMKE, while we have explored the motivating questions for the proposed augmentation strategy, executed experiments open new questions, particularly about the parametrization of Landmark Exchange. We have not found yet found a closed understanding of the

impact of ratio in performance, and in particular the contrast in observations between Domain Generalization and Intra-Dataset benchmarks hints at a potential for further development in other intra-dataset benchmarks, in particular the other three Oulu-NPU ones (Boulkenafet et al., 2017). In general, we have found similar performance between Landmark Exchange and Patch Exchange, contrary to our our initial hypothesis of Landmark Exchange showing better results. For the intra-dataset evaluation on Oulu-NPU protocol 1, however, we did find Landmark Exchange to improve on Patch Exchange's error rates.

Regarding our proposed unbiased DG-FAS benchmarks, we have successfully observed our hypothesis and shown that the bias in current benchmarks is detrimental to model development. Our unbiased alternatives were also shown to better represent real-world model performance. Since we expected the research community to be interested in continuing to use certain datasets (CASIA- FASD, Replay-Attack, MSU-MFSD, and Oulu-NPU) as test sets in DG-FAS benchmarks, we believe the proposed benchmarks will be better accepted in the added validation set variant. We experimented with two different possible datasets for the "added validation", SiW and Rose-Youtu, and reached similar general behavior with both of them. However, since we believe that the data domain of the SiW dataset is related to the ones of the four currently used datasets, it could be more interesting to focus on using SiW instead of Rose-Youtu for validation. Still, we did not find any issues with using Rose-Youtu, and when possible we encourage researchers to use the two different validation dataset possibilities (in separate experiments) for evaluation. In summary, when possible we recommend experimenting with all the unbiased benchmarks proposed in this work. For a more concise battery of experiments, we suggest using the unbiased benchmarks with SiW as validation.

# 6 CONCLUSION

The trajectory of the state of the art in face liveness detection has been thoroughly studied and summarized in this work. Major datasets have also been listed, and the current challenges in the field were discussed. From an understanding of these challenges a novel approach was proposed, namely performing data augmentation with the Landmark Exchange strategy. Presented experiments validate this approach can be effective but highlight that there is still unexplored nuance which is very important in correctly applying both Landmark Exchange and Patch Exchange to model training.

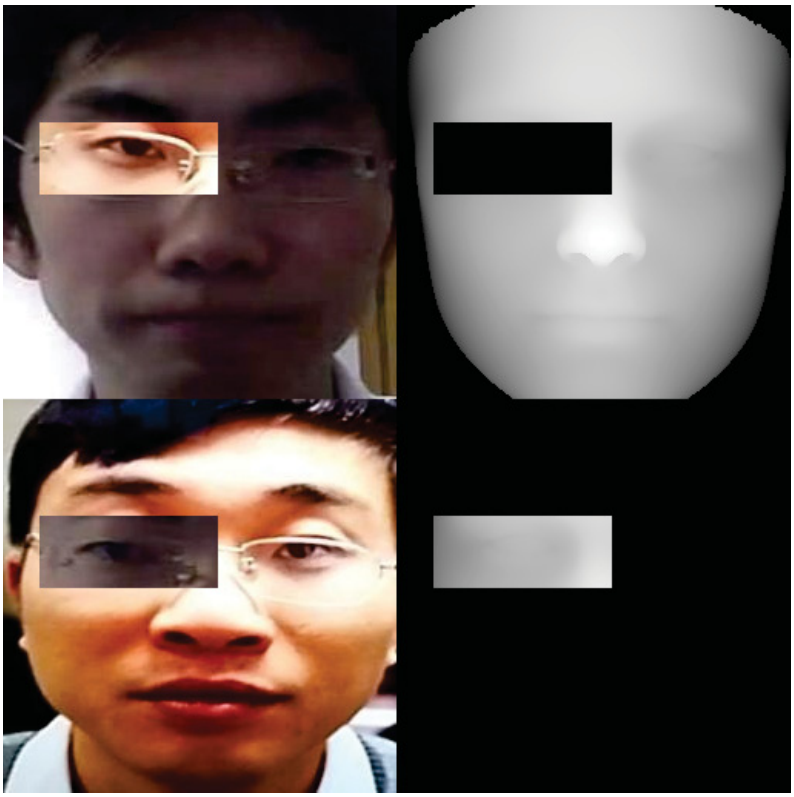


Figura 6.1: Illustration of rectangular exchanges centered on landmarks (in this case, the right eye region) between two images (upper and bottom left) and their corresponding depth maps (upper and bottom right).

Possibilities for future work regarding Landmark Exchange include studying variations that can be used to find middle ground between Landmark and Patch Exchange; for example, exchanging randomly sized rectangular regions centered in face landmarks (see Figure 6.1) instead of the face landmark polygons could be a middle term between the benefits of both augmentation strategies and better affect model performance. We have also not explored the parametrization of these augmentations in the proposed scenarios, i.e., how changing the size and amount of exchanges between a pair of samples could affect results.

We highlighted the bias issue in current DG-FAS evaluation benchmarks and how it negatively affects practice and research. As an alternative, we successfully proposed a benchmark with validation datasets which, with two different suggested datasets, demonstrates the expected behavior of being more challenging than its biased counterpart, which in some cases directly yielded a better test set performance than in previous biased benchmarks. The proposed benchmark should not be difficult to implement in any current code base. We also

executed similar experiments using the WFAS dataset as a test, which is a different task than DG-FAS (introducing new attack types in the test set), and show how important it is to consider more variability in real-world data when designing DG-FAS methods.

We also encourage the exploration of alternative validation datasets and new formats for model evaluation in DG-FAS. By introducing more rigor in research, the field as a whole benefits from consistent, meaningful conclusions that move general knowledge forward. In future work, we intend to further improve current benchmarks. Of particular interest are two questions regarding the WFAS dataset: (1) can we improve DG-FAS models to have better performance in WFAS? And (2) can we use the best-performing models at WFAS in the DG-FAS task? We believe these are relevant tasks because of the particular characteristics of WFAS.

# REFERÊNCIAS

30107-3:2023, I. (2023). Information technology — Biometric presentation attack detection. Standard, International Organization for Standardization, Geneva, CH.

Agarwal, A., Yadav, D., Kohli, N., Singh, R., Vatsa, M. e Noore, A. (2017). Face presentation attack with latex masks in multispectral videos. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Agência Senado (2024). Golpes digitais atingem 24% da população brasileira, revela DataSenado. https://www12.senado.leg.br/noticias/materias/2024/10/01/golpes-digitais-atingem-24-da-populacao-brasileira-revela-datasenado.

Anjos, A. e Marcel, S. (2011). Counter-measures to photo attacks in face recognition: A public database and a baseline. Em *2011 International Joint Conference on Biometrics (IJCB)*, páginas 1–7.

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

Boulkenafet, Z., Komulainen, J. e Hadid, A. (2016). Face spoofing detection using colour texture analysis. *IEEE Transactions on Information Forensics and Security*, 11(8):1818–1830.

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

Chan, P. P. K., Liu, W., Chen, D., Yeung, D. S., Zhang, F., Wang, X. e Hsu, C.-C. (2018). Face liveness detection using a flash against 2d spoofing attack. *IEEE Transactions on Information Forensics and Security*, 13(2):521–534.

Chen, B., Yang, W., Li, H., Wang, S. e Kwong, S. (2021). Camera invariant feature learning for generalized face anti-spoofing. *IEEE Transactions on Information Forensics and Security*, 16:2477–2492.

Chen, H., Hu, G., Lei, Z., Chen, Y., Robertson, N. M. e Li, S. Z. (2020). Attention-based two-stream convolutional networks for face spoofing detection. *IEEE Transactions on Information Forensics and Security*, 15:578–593.

Chingovska, I., Anjos, A. e Marcel, S. (2012). On the effectiveness of local binary patterns in face anti-spoofing. Em *2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*, páginas 1–7.

Chingovska, I., Anjos, A. R. d. e Marcel, S. (2014). Biometrics evaluation under spoofing attacks. *IEEE Transactions on Information Forensics and Security*, 9(12):2264–2276.

Cho, W., Choi, S., Park, D. K., Shin, I. e Choo, J. (2019). Image-to-image translation via group-wise deep whitening-and-coloring transformation.

de Freitas Pereira, T., Anjos, A., De Martino, J. M. e Marcel, S. (2013). Lbp-top based countermeasure against face spoofing attacks. Em Park, J.-I. e Kim, J., editores, *Computer Vision - ACCV 2012 Workshops*, páginas 121–132, Berlin, Heidelberg. Springer Berlin Heidelberg.

Deb, D. e Jain, A. K. (2021). Look locally infer globally: A generalizable face anti-spoofing approach. *IEEE Transactions on Information Forensics and Security*, 16:1143–1157.

Dumoulin, V., Shlens, J. e Kudlur, M. (2017). A learned representation for artistic style.

Galbally, J., Alonso-Fernandez, F., Fierrez, J. e Ortega-Garcia, J. (2012). A high performance fingerprint liveness detection method based on quality related features. *Future Generation Computer Systems*, 28(1):311–321.

Garg, S., Mittal, S., Kumar, P. e Anant Athavale, V. (2020). Debnet: Multilayer deep network for liveness detection in face recognition system. Em *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, páginas 1136–1141.

George, A. e Marcel, S. (2021). On the effectiveness of vision transformers for zero-shot face anti-spoofing. Em *2021 IEEE International Joint Conference on Biometrics (IJCB)*, páginas 1–8.

Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z. e Li, S. Z. (2020). Towards fast, accurate and stable 3d dense face alignment. Em *Proceedings of the European Conference on Computer Vision (ECCV)*.

Hanley, J. A. e McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36. PMID: 7063747.

He, K., Zhang, X., Ren, S. e Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Heusch, G., George, A., Geissbühler, D., Mostaani, Z. e Marcel, S. (2020). Deep models and shortwave infrared information to detect face presentation attacks. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(4):399–409.

Huang, X. e Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization.

Ito, K., Okano, T. e Aoki, T. (2017). Recent advances in biometrie security: A case study of liveness detection in face recognition. Em *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, páginas 220–227.

Jia, Y., Zhang, J., Shan, S. e Chen, X. (2020). Single-side domain generalization for face anti-spoofing. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jourabloo, A., Liu, Y. e Liu, X. (2018). Face de-spoofing: Anti-spoofing via noise modeling. Em *ECCV*.

Killioğlu, M., Taşkiran, M. e Kahraman, N. (2017). Anti-spoofing in face recognition with liveness detection using pupil tracking. Em *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, páginas 000087–000092.

Komulainen, J., Hadid, A. e Pietikäinen, M. (2013). Context based face anti-spoofing. Em *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, páginas 1–8.

Koshy, R. e Mahmood, A. (2019). Optimizing deep cnn architectures for face liveness detection. *Entropy*, 21(4).

Le, B. M. e Woo, S. S. (2024). Gradient alignment for cross-domain face anti-spoofing.

Li, H., Li, W., Cao, H., Wang, S., Huang, F. e Kot, A. C. (2018). Unsupervised domain adaptation for face anti-spoofing. *IEEE Transactions on Information Forensics and Security*, 13(7):1794–1809.

Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M. e Hadid, A. (2016). An original face anti-spoofing approach using partial convolutional neural network. Em *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, páginas 1–6.

Li, L., Xia, Z., Hadid, A., Jiang, X., Zhang, H. e Feng, X. (2019). Replayed video attack detection based on motion blur analysis. *IEEE Transactions on Information Forensics and Security*, 14(9):2246–2261.

Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X. e Yang, M.-H. (2017). Universal style transfer via feature transforms.

Liu, A., Tan, Z., Wan, J., Liang, Y., Lei, Z., Guo, G. e Li, S. Z. (2021a). Face anti-spoofing via adversarial cross-modality translation. *IEEE Transactions on Information Forensics and Security*, 16:2759–2772.

Liu, A., Wan, J., Escalera, S., Jair Escalante, H., Tan, Z., Yuan, Q., Wang, K., Lin, C., Guo, G., Guyon, I. e Li, S. Z. (2019a). Multi-modal face anti-spoofing attack detection challenge at cvpr2019. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Liu, S., Song, Y., Zhang, M., Zhao, J., Yang, S. e Hou, K. (2019b). An identity authentication method combining liveness detection and face recognition. *Sensors*, 19(21).

Liu, W., Wei, X., Lei, T., Wang, X., Meng, H. e Nandi, A. K. (2021b). Data fusion based two-stage cascade framework for multi-modality face anti-spoofing. *IEEE Transactions on Cognitive and Developmental Systems*, páginas 1–1.

Liu, Y., Jourabloo, A. e Liu, X. (2018a). Learning deep models for face anti-spoofing: Binary or auxiliary supervision. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y., Jourabloo, A. e Liu, X. (2018b). Learning deep models for face anti-spoofing: Binary or auxiliary supervision.

Liu, Y., Stehouwer, J., Jourabloo, A. e Liu, X. (2019c). Deep tree learning for zero-shot face anti-spoofing. Em *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 4675–4684.

Luo, S., Kan, M., Wu, S., Chen, X. e Shan, S. (2018). Face anti-spoofing with multi-scale information. Em *2018 24th International Conference on Pattern Recognition (ICPR)*, páginas 3402–3407.

Mostaani, Z., George, A., Heusch, G., Geissbühler, D. e Marcel, S. (2020). The high-quality wide multi-channel attack (HQ-WMCA) database. *CoRR*, abs/2009.09703.

Patel, K., Han, H. e Jain, A. K. (2016). Secure face unlock: Spoof detection on smartphones. *IEEE Transactions on Information Forensics and Security*, 11(10):2268–2283.

Purnapatra, S., Smalt, N., Bahmani, K., Das, P., Yambay, D., Mohammadi, A., George, A., Bourlai, T., Marcel, S., Schuckers, S., Fang, M., Damer, N., Boutros, F., Kuijper, A., Kantarci, A., Demir, B., Yildiz, Z., Ghafoory, Z., Dertli, H., Ekenel, H. K., Vu, S., Christophides, V., Dashuang, L., Guanghao, Z., Zhanlong, H., Junfu, L., Yufeng, J., Liu, S., Huang, S., Kuei, S., Singh, J. M. e Ramachandra, R. (2021). Face liveness detection competition (livdet-face) - 2021. Em *2021 IEEE International Joint Conference on Biometrics (IJCB)*, páginas 1–10.

Quan, R., Wu, Y., Yu, X. e Yang, Y. (2021). Progressive transfer learning for face anti-spoofing. *IEEE Transactions on Image Processing*, 30:3946–3955.

Ramachandra, R. e Busch, C. (2017). Presentation attack detection methods for face recognition systems: A comprehensive survey. *ACM Comput. Surv.*, 50(1).

Roy, S., Siarohin, A., Sangineto, E., Bulo, S. R., Sebe, N. e Ricci, E. (2020). Unsupervised domain adaptation using feature-whitening and consensus loss.

Sanghvi, N., Singh, S. K., Agarwal, A., Vatsa, M. e Singh, R. (2021). Mixnet for generalized face presentation attack detection. Em *2020 25th International Conference on Pattern Recognition (ICPR)*, páginas 5511–5518.

Shao, R., Lan, X., Li, J. e Yuen, P. C. (2019). Multi-adversarial discriminative deep domain generalization for face presentation attack detection. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shen, T., Huang, Y. e Tong, Z. (2019). Facebagnet: Bag-of-local-features model for multi-modal face anti-spoofing. Em *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, páginas 1611–1616.

Singh, M. e Arora, A. S. (2018). A novel face liveness detection algorithm with multiple liveness indicators. *Wireless Personal Communications*, 100(4):1677–1687.

Sun, B. e Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation.

Sun, Y., Liu, Y., Liu, X., Li, Y. e Chu, W.-S. (2023). Rethinking domain generalization for face anti-spoofing: Separability and alignment.

Tan, X., Li, Y., Liu, J. e Jiang, L. (2010). Face liveness detection from a single image with sparse low rank bilinear discriminative model. Em Daniilidis, K., Maragos, P. e Paragios, N., editores, *Computer Vision – ECCV 2010*, páginas 504–517, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wang, D., Guo, J., Shao, Q., He, H., Chen, Z., Xiao, C., Liu, A., Escalera, S., Escalante, H. J., Lei, Z., Wan, J. e Deng, J. (2023). Wild face anti-spoofing challenge 2023: Benchmark and results.

Wang, G., Han, H., Shan, S. e Chen, X. (2020a). Cross-domain face presentation attack detection via multi-domain disentangled representation learning. Em *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 6677–6686.

Wang, G., Lan, C., Han, H., Shan, S. e Chen, X. (2019). Multi-modal face presentation attack detection via spatial and channel attentions. Em *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, páginas 1584–1590.

Wang, J., Zhang, J., Bian, Y., Cai, Y., Wang, C. e Pu, S. (2021a). Self-domain adaptation for face anti-spoofing. *CoRR*, abs/2102.12129.

Wang, Y., Song, X., Xu, T., Feng, Z. e Wu, X.-J. (2021b). From rgb to depth: Domain transfer network for face anti-spoofing. *IEEE Transactions on Information Forensics and Security*, 16:4280–4290.

Wang, Z., Wang, Z., Yu, Z., Deng, W., Li, J., Gao, T. e Wang, Z. (2022). Domain generalization via shuffled style assembly for face anti-spoofing.

Wang, Z., Yu, Z., Zhao, C., Zhu, X., Qin, Y., Zhou, Q., Zhou, F. e Lei, Z. (2020b). Deep spatial gradient and temporal depth learning for face anti-spoofing. Em *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 5041–5050.

Wen, D., Han, H. e Jain, A. K. (2015). Face spoof detection with image distortion analysis. *IEEE Transactions on Information Forensics and Security*, 10(4):746–761.

Wightman, R. (2019). Pytorch image models. https://github.com/rwightman/pytorch-image-models.

Yang, J., Lei, Z., Yi, D. e Li, S. Z. (2015). Person-specific face antispoofing with subject domain adaptation. *IEEE Transactions on Information Forensics and Security*, 10(4):797–809.

Yang, X., Luo, W., Bao, L., Gao, Y., Gong, D., Zheng, S., Li, Z. e Liu, W. (2019). Face anti-spoofing: Model matters, so does data. Em *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 3502–3511.

Yu, Z., Qin, Y., Li, X., Zhao, C., Lei, Z. e Zhao, G. (2022). Deep learning for face anti-spoofing: A survey.

Yu, Z., Qin, Y., Zhao, H., Li, X. e Zhao, G. (2021). Dual-cross central difference network for face anti-spoofing.

Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., Zhou, F. e Zhao, G. (2020a). Searching central difference convolutional networks for face anti-spoofing.

Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., Zhou, F. e Zhao, G. (2020b). Searching central difference convolutional networks for face anti-spoofing. Em *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 5294–5304.

Zhang, K.-Y., Yao, T., Zhang, J., Tai, Y., Ding, S., Li, J., Huang, F., Song, H. e Ma, L. (2020). Face anti-spoofing via disentangled representation learning. Em *European Conference on Computer Vision*, páginas 641–657. Springer.

Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D. e Li, S. Z. (2012). A face antispoofing database with diverse attacks. Em *2012 5th IAPR International Conference on Biometrics (ICB)*, páginas 26–31.

Zheng, W., Yue, M., Zhao, S. e Liu, S. (2021). Attention-based spatial-temporal multi-scale network for face anti-spoofing. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(3):296–307.

Zhou, Q., Zhang, K.-Y., Yao, T., Lu, X., Yi, R., Ding, S. e Ma, L. (2023). Instance-aware domain generalization for face anti-spoofing. Em *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.