Multi-adversarial Domain Generalization to Improve Face Recognition Reliability

Daniel Prazeres Baptista
daniel.p.baptista@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2021

Abstract

Facial recognition is one of the most popular technologies nowadays, constituting the first security barrier for devices like smartphones and tablets. This, in turn, makes facial recognition systems vulnerable to attacks with one of the most notorious being face presentation attacks. Face presentation attacks are an emerging threat and therefore have become more complex and unpredictable, through the years. The challenge of detecting face presentation attacks has led to the appearance of solutions based on liveness detection, facial appearance, contextual information, and more recently, solutions based on deep learning techniques. Within the deep learning field, one topic that has been explored to recognize such attacks is the domain generalization topic. This work focuses on a solution that incorporates this topic. The adopted approach improves on an existing solution, taken as baseline, that trains a model with face presentation attacks seen in different conditions, to be able to generalize to other acquisition conditions. The present work proposes a source domain reorganization to enhance the generalized feature space, together with a modified triplet loss function that is more suitable for the proposed domain reorganization. The experiments were conducted on four public datasets. The solution proposed includes domain reorganization and a more suitable triplet loss function, achieving on-pair performance when tested with REPLAY-ATTACK dataset and outperforming the baseline architecture in the CASIA and MSU datasets. Future work includes complementing the proposed solution with a liveness detection algorithm, and a solution for addressing previously unseen attacks.
Keywords: Face presentation attack detection, domain generalization, multiple source domains, triplet loss, remote photoplethysmography

1. Introduction

This section presents the problem's motivation, related work and the contributions of the proposed work.

1.1. Motivation

Facial recognition systems rely on the uniqueness of a person's facial features to recognize an individual. Without the burden of having to memorize a password or carrying a card, focusing on an intrinsic biometric property of an individual (the person's face), provides a solution of improved security.

With the widespread usage of cameras and webcams in recent years, accompanied by the constant need to have users confidentiality preserved, utilizing face biometric data as an access key, has opened new application areas. The widespread use of face recognition systems makes them a target for attacks, notably to allow an attacker to impersonate a genuine user. A presentation attack (PA), also known as a spoofing attack, that for example, can be as simple as presenting to the system a non-living spoof, or a disguise, known as presentation attack instruments (PAI), to look like someone else, or to hide a person's own identity.

There is a range of possible PAs, including: printed photos, videos and photos displayed on the screen of portable devices, face masks, make-up and, in extreme scenarios, plastic surgery.

In order to mitigate the consequences of presentation attacks, finding effective ways to fight impostor attempts to spoof biometric systems are becoming urgent. By detecting the presence of a living body, any type of objects with the goal of scamming the system will be detected and consequently the access to the impostor will be denied.

1.2. Related work

PAD technology. Presentation attack detection (PAD) methods can be categorized as follows:

- Liveness detection - The main goal of liveness detection methods is to identify physiological signs of life. These proofs of liveness can be provided by an interaction with the user, requiring his cooperation or not (voluntary or involuntary, respectively), which in this case head movement detection [16, 2], blink detection [31], challenge-response [1, 30] have been proposed, and/or using techniques such as remote photoplethysmography (rPPG) to detect the presence of a heart rate [22, 27].

- Facial appearance - Methods that use image properties to detect PAs. In some cases these methods can take in account temporal information, for instance to detect video replay attacks, or they may be designed to work on individual images. This type of methods includes frequency techniques [21, 23, 32, 4, 9], texture analysis based methods [26, 20, 5], image quality assessment (IQA) [12, 13, 33] and motion based methods [17, 28].

- Contextual information - Methods exploring background information to detect PAs. In some attacks, it is possible to observe suspect content when looking away from the facial region, for example when the impostor presents a printed photo or a display in front of the camera. In these types of methods, contextual scenic cues can contribute with valuable information about the possibility of a PA [18, 19].

To address the diversity of PAs to which a biometric recognition system can be subjected to, multi-modal systems (combining different biometric cues) have been proposed as a promising solution to this problem [24, 10].

Zero-shot learning. The appearance of samples from unseen classes is a continuous problem in the PAD context, since the variety of PAs carried out by attackers is immense and constantly evolving. To overcome this issue, zero-shot learning (ZSL) appeared as a solution to PAD by learning generalized and discriminative features from a set of known PAs for unseen novel PAs [25]. Collecting labeled data for every new attack is impossible, so ZSL tries to be able of detecting novelty attack types, while not having samples from these attacks on the training set.

Domain generalization. Domain generalization makes the assumption that a generalized feature space exists that the multiple source domains and the unseen target domain have in common, which enables generalization capability to unseen domains [29, 15]. In contrast to ZSL, domain generalization focuses on the PAD problem by having training and testing data with the same types of attacks but obtained in different conditions (PAI, illumination, background, devices,...), which translates to having training and test data from the same classes but with different distributions.

1.3. Contributions

The work basis was a domain generalization solution, that from a set of domains sharing different facial image acquisition conditions, and using the auxiliary cues obtained from a triplet loss function and a depth estimator, tries to learn a shared feature space able to distinguish real and fake faces. In an attempt to make this solution more robust and provide better classification results when being tested, a set of innovative contributions was implemented, and additional ideas to be pursued in future research are proposed:

- **Domain reorganization** – This proposal consists in a reorganization of the datasets/domains used for training the system, using an attack-oriented organization, to extract more reliable generalization cues relying on the characteristics shared between PA types.

- **Triplet loss function modification** – This proposal consists in a modification of the baseline triplet loss function, to better learn how to separate the different attacks types in the feature space, while clustering real faces closer together.

- **Incorporation of rPPG** – This is a proposal for further work, as the conducted tests still need to be extended. It consists in combining the baseline solution with a different strategy, able to detect the heart rate pulse from facial images. It is therefore the proposal for a multiple cue approach, to achieve a more robust PAD solution, able to improve classification results.

2. Domain generalization baseline

This section introduces the method that inspired the work developed, entitled "Multi-adversarial Discriminative Deep Domain Generalization for Face Presentation Attack Detection" [29].

2.1. Baseline architecture

The baseline method,presented in [29] had the objective of learning a generalized feature space, capable of identifying PAs obtained in conditions that are different from the ones observed during training. This method is composed by three main components: (i) multi-adversarial domain generalization, (ii) triplet loss function, and (iii) depth estimation. The general architecture of this solution is presented in Figure 1.

2.2. Multi-adversarial domain generalization

The multi-adversarial domain generalization (presented with more detail in Figure 2) is divided into two steps: (i) pre-training the feature extractors (M_1,M_2,M_3 in Figure 2) one for each domain;

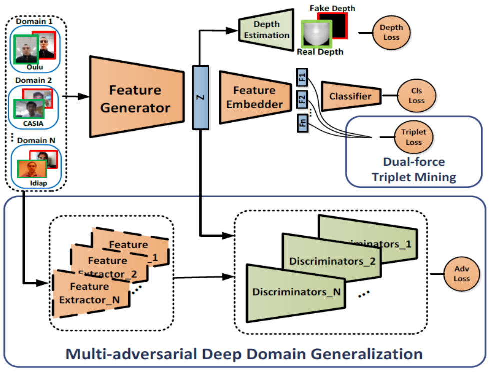

Figure 1: Multi-adversarial deep domain generalization for face PAD: Architecture [29]

(ii) train one feature generator to compete with all the domain discriminators at the same time. The first step obtains a set of discriminative feature spaces, one from each dataset (or domain, in the original paper's terminology), that are biased towards the dataset that originated it, making it unsuitable for generalization to attacks obtained in different conditions. With that in mind, the multi-adversarial implementation tries to create a common feature space, that is sufficiently generic to represent the cases seen in all the the considered source datasets, thus creating a generalized feature space. Using a GAN the generator (denoted as G in Figure 2) tries to learn a generalized feature space capable of simultaneously fooling the various domain discriminators (denoted as D_1, D_2 and D_3 in Figure 2), while each domain discriminator tries to distinguish between the generalized feature space and the respective discriminative feature space.

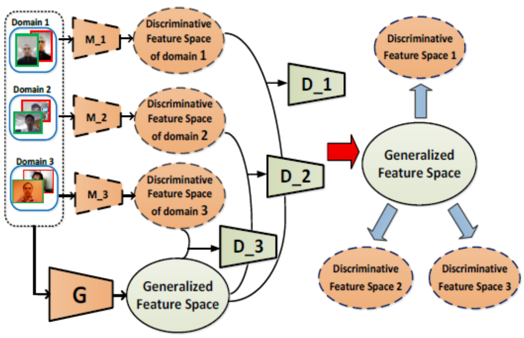

Figure 2: Detailed architecture of multi-adversarial domain generalization [29]

2.3. Triplet loss function

The triplet loss principle can be interpreted with the help of Figure 3. Before applying the triplet-mining constraint, one can assume that presentation attacks and *bonafide* faces from the same individuals share similar characteristics, while PAs and *bonafide* faces from different individuals have different facial characteristics. To address this issue, a triplet loss based constraint

$$
\begin{aligned}
&\mathcal{L}_{Triplet}(X, Y; G, E) = \\
&= \sum_{\substack{a,p,n \\ \forall y_a = y_p \neq y_n, i=j}} [\|E(G(x_i^a)) - E(G(x_j^p))\|_2^2 - \\
&\quad - \|E(G(x_i^a)) - E(G(x_j^n))\|_2^2 + \alpha_1] + \\
&+ \gamma \sum_{\substack{a,p,n \\ \forall y_a = y_p \neq y_n, i \neq j}} [\|E(G(x_i^a)) - E(G(x_k^p))\|_2^2 - \\
&\quad - \|E(G(x_i^a)) - E(G(x_k^n))\|_2^2 + \alpha_2]
\end{aligned} \tag{1}
$$

is designed to: (i) reduce the distance (in the feature space) of each subject sample to its intra-domain positive samples (same dataset) in comparison to the distance to its intra-domain negative samples; and (ii) reduce the distance of each subject sample to its inter-domain positive samples (different dataset) in comparison to the distance to its inter-domain negative samples.

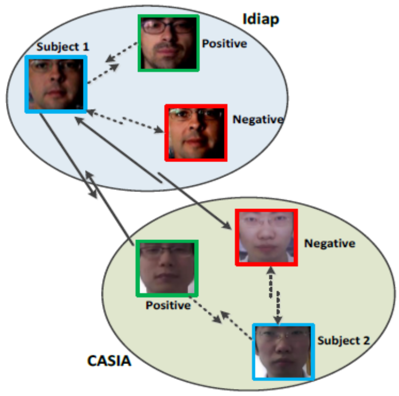

Figure 3: Dual-force triplet-mining constraint objective [29]

2.4. Depth estimation

Depth estimation relies on the fact that *bonafide* faces have depth, while several types of presentation attacks, like photo attacks or video replay attacks, are presented using planar surfaces. The depth information is exploited by measuring the difference between the depth estimated from the output of the feature generator and the ground truth depth. A example of a ground-truth sample is in Figure 4. This information is incorporated since it is possible that the computation of depth information for a given dataset/domain is biased, being included in the learning process to exploit differentiation cues in the generalized feature space.

3. Methodology

The modifications to the baseline method are explained in this section.

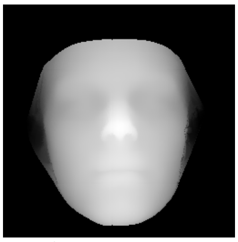

Figure 4: Example of depth estimation using the PRNet software[11]

Table 1: Table with the organization of the domains for each training case

Training	Domain 1 samples	Domain 2 samples	Domain 3 samples	Testing
MSU REPLAY ATTACK OULU	Real(all) Print attack(all)	Real(all) Replay attack (all samples from REPLAY-ATTACK and OULU; phone as PAI, from MSU)	Real(all) Replay attack (tablet as PAI, from MSU only)	CASIA
MSU OULU CASIA	Real(all) Print attack (all from MSU and OULU; warped/flat only, from CASIA)	Real(all) Replay attack(all)	Real(all) Eye-cut print attack (all from CASIA)	REPLAY ATTACK
	Real(all) Print attack(all)	Real(all) Replay attack (all samples from CASIA and OULU; using phone as PAI, from MSU)	Real(all) Replay attack (tablet as PAI, from MSU only)	
MSU REPLAY ATTACK CASIA	Real(all) Print attack (all from MSU and OULU; warped/flat only from CASIA)	Real(all) Replay attack(all)	Real(all) Eye-cut print attack (all from CASIA)	OULU
	Real(all) Print attack(all)	Real(all) Replay attack (all from CASIA and REPLAY; using phone as PAI, from MSU)	Real(all) Replay attack (tablet as PAI; from MSU only)	
REPLAY ATTACK OULU CASIA	Real(all) Print attack (all from REPLAY and OULU; warped/flat only, from CASIA)	Real(all) Replay attack (all)	Real(all) Eye-cut print attack (all from CASIA)	MSU

3.1. Proposal 1: Modifying the domain generalization procedure

As mentioned before, the organization of the data into domains plays a crucial role on learning a generalized model and therefore also on the discriminative power of the obtained features.

In the baseline solution, each dataset is allocated to a domain, representing a set of conditions shared by the images of that dataset. In this proposal, depending on the datasets used for training, an alternative domain constitution is proposed, using the type of PA as criteria for grouping the available data into the relevant domains.

Among the available datasets, for IDIAP REPLAY-ATTACK and OULU datasets the available PA types were organized into: (i) print, and (ii) replay attacks. For the remaining datasets used for training, notably CASIA and MSU, the division is more elaborated. CASIA is organized into: (i) flat/warped prints, (ii) eye-cut prints, and (iii) replay attacks. On the other hand, for MSU the attacks are organized as: (i) print attacks, (ii) replay attacks on phones and (iii) replay attacks on tablets. Each domain will have samples from *bonafide* and one type of PA extracted from all the databases considered for training. The training cases are in Table 1.

The default dataset division, to organize source domains for training, requires separating genuine, print and replay attacks. Since the considered architecture considers three source domains, when CASIA and MSU are both used for training, only one of them can be divided as supposed, as the CASIA dataset print attack samples are separated into eye-cut print attacks and attacks using flat and warped paper. For the MSU dataset, replay attacks can be separated into attacks obtained with tablets and phones. To able to consider the above divisions of the CASIA and MSU datasets, the architecture would need to consider more than three source domains. Therefore, when testing with the OULU and REPLAY-ATTACK datasets there are two possibilities: (i) dividing CASIA print attacks (flat/warped & cut) while for MSU all replay attacks are considered as a single type of attack; (ii) dividing MSU replay attacks according to the PAI used (tablet and phone) while for the CASIA dataset all print attacks are considered together.

With this arrangement, the focus of the model is to identify discriminative features based on the characteristics shared by each type of attack, whatever the acquisition conditions, rather than focusing on the specific conditions found within each dataset, as considered by the baseline architecture. The domain organization considered by this proposal is illustrated in 5.

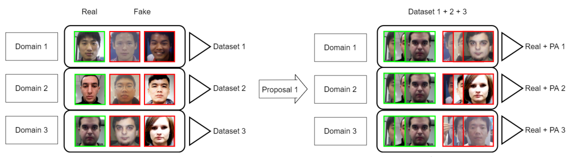

Figure 5: Baseline organization and the modifications that led to proposal 1

3.2. Proposal 2: Modifying the domain generalization procedure and the triplet loss function

The second proposal made in this dissertation is to consider an additional modification on top of what was proposed in 3.1. It consist in modifying the triplet loss function, still with the goal to minimize intra-class distance while maximizing inter-class distance in both intra and cross domains, but now considering a function that is more in line with the new organization of the training data into domains (which no longer include samples from a single dataset). Since, the proposed modification regarding source domain organization focuses on separating domains by PA attack types, the modified triplet loss function follows the same logic. Therefore, the triplet loss function should minimize intra-class distance while maximizing inter-class distance, resulting in a separation of the different PA and the aggregation of real samples. The implementation of this solution follows the same steps of 3.1 with the exception of the triplet loss function, which instead of Equation 1, now uses

Equation 2, removing the cross-domain term:

$$\mathcal{L}_{Triplet}(X, Y; G, E) = \sum_{\substack{a,p,n \\ \forall y_a = y_p \neq y_n, i=j}} \|E(G(x_i^a)) - E(G(x_j^p))\|_2^2 - \\ - \|E(G(x_i^a)) - E(G(x_j^n))\|_2^2 + \alpha_1 \quad (2)$$

3.3. Proposal 3: Adding Remote Photoplethysmography

In this proposal, the focus is not on the domain generalization capacity of the solution, but instead in considering an additional cue, rPPG, to improve the classification results. Therefore, including the rPPG module doesn't aim at improving the domain generalization capability, but rather to enhance the overall classification performance of resulting system.

The system proposed here consists of combining the baseline architecture, or one of the modified proposals presented in the previous sections, with an rPPG module. A possible architecture for the resulting system is proposed in 6.

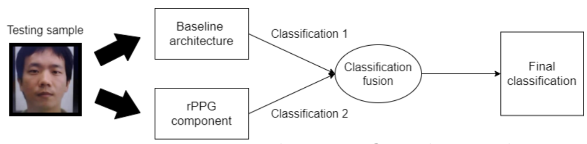

Figure 6: Proposal 3 modification: Classification fusion of baseline and rPPG classifications

To use rPPG methods that estimate the heart rate for PAD, two additional steps are considered in Figure 7: (i) gathering a feature vector with statistical information of the heart rate estimation along the video duration (blocks highlighted in green in the figure); and (ii) applying a classifier, such as SVM, to obtain a spoof/real score (blocks with white background in the figure).

4. Experimental results and discussion

In this section, the experimental setup and datasets used to obtain the experimental results are presented. Next, the results for the baseline and each proposal are reported.

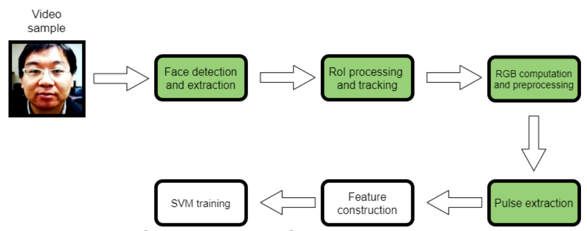

Figure 7: rPPG component: Step-by-step design of proposal three, with green blocks representing the steps to estimate heart rate and white blocks the steps for the classification task

4.1. Experimental setup and datasets

To be able to obtain experimental results, the images of four datasets will be needed in the two phases comprising the experimental process: training and testing. Three datasets have their examples exclusively distributed across the source domains to perform training and the examples of the remaining dataset are used for testing. Each of the three datasets used for training contributes with all their samples and, in the testing phase, classification is performed using all the samples of the test dataset. The distribution of samples considered for training the baseline and the proposed solutions were presented.

The training environment used was Google Colaboratory [14]. This environment offers free access to GPU (indispensable to run any deep neural network) but with usage limitation, which led to the training of the baseline and of the proposed solutions to need some adjustments, notably: (i) decrease the batch size; and (ii) limit the maximum number of epochs when training a model. The original implementation of the baseline solution used a batch size of 20 per domain [29], while for the re-implemented version and for implementation of the modification proposals a batch size of 3 had to be considered. The size of the datasets used for training, which was different for the various evaluation scenarios considered, had a direct impact on the number of epochs completed due to usage limitations, culminating in different values across the various scenarios.

Before proceeding to the training phase, two more conditions need to be defined: (i) the optimizer needs to be chosen and configured, and (ii) the hyperparameters of the triplet loss function need to be set.

The optimizer used was Adam. The learning rates for the two phases of training are: 10^{-5} for the first phase, which consists in training the generator, embedder, classifier and discriminators together, and the second phase where the generator and depth estimator are trained simultaneously with learning rate 10^{-4}. β_1 and β_2 are equal to 0.9 and 0.999, respectively. ϵ maintains the default value of 10^{-8}.

Thus, the hyperparameters γ, α_1, and α_2, involved in the triplet loss function defined in 1, are set to 0.1, 0.1, and 0.5, respectively. These values were reported in the original article as the ones used on the main solution and therefore, were adopted for all the experimental cases.

In the testing phase, given a certain sample, the classifier outputs a score corresponding to the probability of that sample being genuine. This means that higher scores correspond to higher probabilities of the input image coming from a real

Table 2: Summary: Datasets

Name	Year	Subjects	PA type(s)	Genuine & PAs samples
CASIA-FASD	2012	50	Print(flat,warped,cut); Replay(tablet)	150/450
REPLAY-ATTACK	2012	50	Print(flat); Replay(tablet, phone)	200/1000
MSU-MFSD	2015	35	Print(flat); Replay(tablet, phone)	70/210
OULU-NPU	2017	55	Print(flat); Replay(phone)	1980/3960

face, while lower score values reflect a higher probability of being a PA.

The established threshold to refer to a sample as a FN, FP, TN or TP is 50%. A sample is presumed as an attack (N - negative) if it's classification score is below 50%, and presumed as genuine (P- positive) otherwise. Then, depending on whether the sample label obtained from the classification score matches the ground-truth label or not, it is called true (T) or false (F).

Regarding performance evaluation, the two metrics AUC and HTER are adopted; these metrics, complemented by the classification score, allow comparing performance of the various proposals. It is also possible to compare against the original implementation of the baseline, as the paper proposing the baseline architecture [29] uses the same metrics.

The four public face-antispoofing datasets used to perform training and testing were CASIA-FASD [34], REPLAY-ATTACK [7], MSU-MFSD [33] and OULU-NPU [6]. These datasets are summarized in Table 2.

4.2. Baseline architecture results

The obtained results correspond to exactly the same solution, but they are very different. Comparing these results it is possible to observe that the re-implemented version performed a lot worse for most of the domains/datasets combinations considered. Only for the condition where the tests were performed on the MSU dataset we can observe somewhat similar results. These differences can be explained due to the different amount of resources available for training the model that were available for the present work, which impacted the maximum value of the batch size and the limits for memory usage. As such, the difference between both sets of results reflects the insufficient training in the re-implemented model, which, in most cases, is still far from a convergence situation, preventing to achieve a smaller gap between the two implementations as would be expected.

4.3. Proposal 1 results

Comparing to the baseline, the results obtained were satisfactory, with the tests performed on the CASIA and REPLAY-ATTACK datasets showing some improvements in AUC. In Figure 8 are some genuine examples of CASIA samples that achieve better scores with proposal 1 and in Figure 9 the comparison, regarding *bonafide* samples classification, between training divisions is visualized, in which CASIA division shows better results. On the contrary, the results were not as good with MSU dataset and were poor when testing with the OULU dataset, presenting no improvements in the two metrics.

Table 3: Baseline results: comparison between the results reported in [29] (original) and those obtained with the available computational resources reported on 4.1 (re-implemented)

		Original		Re-implemented	
Training	Testing	AUC(%)	HTER(%)	AUC(%)	HTER(%)
OULU MSU REPLAY-ATTACK	CASIA	84.51	24.50	66.13	39.49
CASIA OULU REPLAY-ATTACK	MSU	88.06	17.69	84.72	22.62
CASIA MSU REPLAY-ATTACK	OULU	80.02	27.98	59.30	43.01
OULU CASIA MSU	REPLAY	84.99	22.19	61.79	37.55

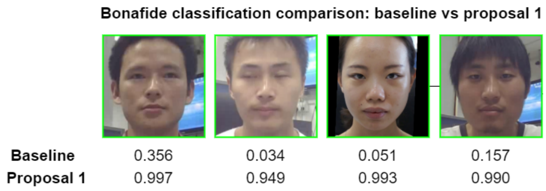

Figure 8: Baseline vs proposal 1: comparing *bonafide* samples classification in CASIA dataset

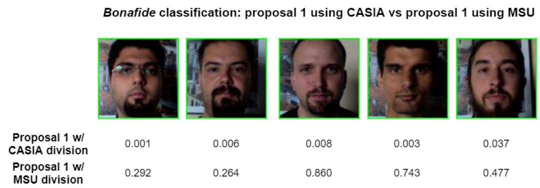

Figure 9: Proposal 1 using CASIA division vs proposal 1 using MSU division: comparing *bonafide* samples classification on REPLAY-ATTACK dataset

4.4. Proposal 2 results

The results obtained for proposal two, that consists in adding to the modification of the previous proposal also a changed triplet loss function, were satisfying in testing with CASIA and MSU datasets, showing improvements in both AUC and HTER metrics. The improvements introduced by this proposal compared to the other implementations are in Figure 10 regarding the MSU dataset, and in Figure 11 improvements compared to the baseline when it came to testing with CASIA dataset. The testing case with OULU dataset, does not achieve an improvement compared to the baseline, but some of the results are very similar. Testing with the REPLAY-ATTACK dataset provides either similar or better results in comparison with the baseline solution. In Figure 12 are some REPLAY-ATTACK genuine samples that were better classified in proposal 2 than in proposal 1.

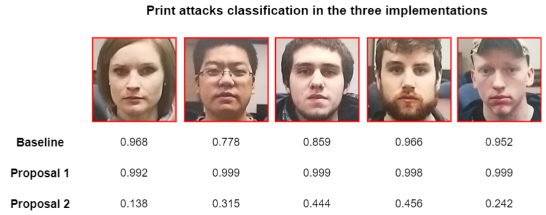

Figure 10: Proposal 2: improvements presented by proposal 2 on print attack classification on MSU dataset

5. Proposal 3 results

The objective of this proposal was to enhance the classification scores, by complementing the domain generalization solution with a different approach, capable of exploring another type of cue, in this case liveness detection, that will hopefully lead to better PA/*bonafide* classification.

For this purpose, reliably detecting the heart rate, measured in beats per minute (BPM), for *bonafide* samples is crucial. For instance, it is known that for an adult the resting heart rate is expected to be between 60 and 100 BPM. And, ideally, no heart rate should be detected for a PA sample. The solutions used to obtain the average BPM value for a given sample, comprehend the steps highlighted in green in Figure 7. The pyVHR[3] solution relies on a set of rPPG algorithms to estimate the heart rate, while the PythonVideoPulserateV2 solution[8] uses a chrominance-based method, focused on improved motion robustness.

With pyVHR it is possible to try a variety of algorithms, and in the context of the present work the goal was to check if any of them adapted well to the datasets used, providing useful information to differentiate between *bonafide* and PAs, through the BPM heart rate estimation provided. Some examples, with the respective BPM value predictions, are presented in Figure 13 and in Figure 14.

6. Discussion

The overall results are summarized in 4.

Proposal 1 achieved better AUC score when testing with CASIA and REPLAY-ATTACK, but not with MSU and OULU. As for HTER, the proposal

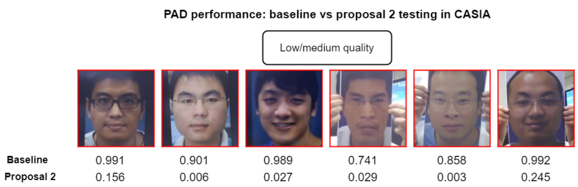

Figure 11: Baseline vs proposal 2: Low and medium attack samples classification comparison in CASIA dataset

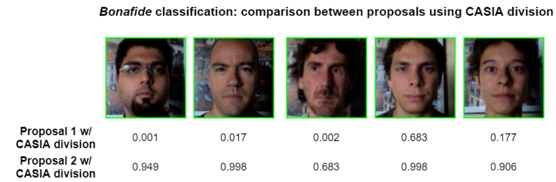

Figure 12: Proposal 1 vs proposal 2: Comparison between proposals 1 and 2 (using CASIA attack division in training) regarding *bonafide* samples of REPLAY-ATTACK dataset

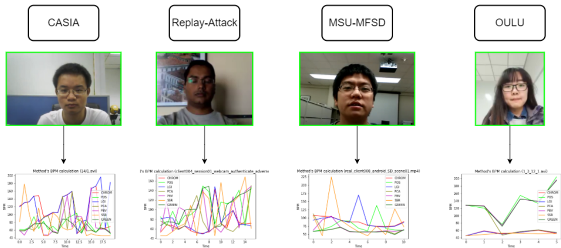

Figure 13: pyVHR[3]: Results on *bonafide* samples

showed no improvements, being only capable of matching the HTER of the baseline architecture with MSU testing.

With proposal 2, the AUC score was superior in CASIA, MSU and in one of the cases of REPLAY-ATTACK, while achieving results similar to the baseline in most of the remaining cases. With this proposal some improvements were also observed in terms of HTER, having CASIA and MSU achieved a better result.

In proposal 3, looking at the results, the choice for the best fitting algorithm is not obvious and most importantly, it seems that none of the algorithms was able to do a satisfying job regarding heart rate prediction with a large sample of images from the considered datasets. For *bonafide* samples, for example, not only the BPM values are sometimes very imprecise, but also, the value variations for different portions of the same video are in some cases very pronounced, which does not correspond to the real situation. The PAs, overall, present a behavior more in line with what was expected - values outside the 60-100 BPM range and with absurd variations. Since is not possible to do a good distinction between the two types of samples, this technique cannot be readily adopted for the desired purpose of detecting PA.

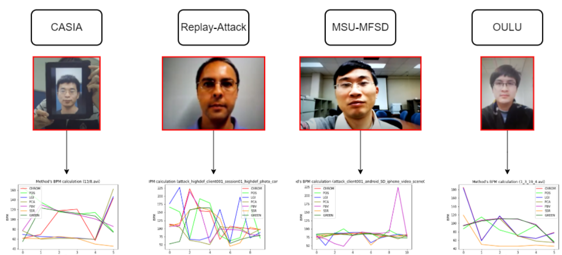

Figure 14: pyVHR[3]: Results on PA samples

Table 4: Experimental results summary: (1) CASIA print attack division (flat/warped & cut); (2) MSU replay attack division (tablet & phone) and in bold the best AUC and HTER for each testing case

Testing	Baseline	Proposal 1	Proposal 2
CASIA	AUC: 66.13 % HTER: 39.49 %	AUC: 69.08 % HTER: 39.71 %	**AUC: 69.11 %** **HTER: 35.29 %**
MSU	AUC: 84.72 % HTER: 22.62 %	AUC: 81.06 % HTER: 22.62 %	**AUC: 87.02 %** **HTER: 21.90 %**
OULU	**AUC: 59.30 %** **HTER: 43.01 %**	AUC: 47.12 % HTER: 51.82 % (1)	AUC: 48.28 % HTER: 52.05 % (1)
		AUC: 49.28 % HTER: 51.00 % (2)	AUC: 57.56 % HTER: 43.31 % (2)
REPLAY-ATTACK	AUC: 61.79 % **HTER: 37.55 %**	AUC: 62.64 % HTER: 40.35 % (1)	**AUC: 67.79 %** HTER: 38.10 % (1)
		AUC: 65.47 % HTER: 39.80 % (2)	AUC: 60.75 % HTER: 39.70 % (2)

7. Conclusions and future work

In this section, the conclusions are drawn and future work is discussed.

7.1. Conclusions

With the goal of improving generalization in PAD, using as baseline a recently published solution [29], this work introduced a group of proposals, in an attempt to collect better generalization cues. The proposals made target two fundamental elements of the solution: the source domain organization considered for training the model, and the triplet loss function used to improve the system classification performance. A third proposal was considered, relying in the incorporation of an heart rate estimation technique, based on rPPG, which was expected to help improve the classification results for cases where the domain generalization would present some limitations. However, the tested rPPG implementations did not perform on the videos available from the databases used in this work.

Proposal two, including both contributions listed above, has achieved interesting results. Of the four tested cases, with different combinations of the datasets used for training and for testing, the proposed solution has achieved better results than the baseline in two of those cases (testing with the CASIA and with the MSU datasets). For the other two cases, each including two sub-cases, the results with the OULU dataset were inferior or similar to the baseline, while tests with the Replay-Attack dataset have shown similar or better results than the baseline. In general, this proposal was able to contribute with promising results, since in most testing cases the metric results are better, and those that are not, are very similar to the baseline.

The improvements observed when compared to the baseline suggest that the proposed modifications are of interest to improve PAD domain generalization. Therefore, it would be desirable to repeat the same tests using a more powerful computational platform, notably including a machine with an appropriate GPU, capable of handling bigger batch sizes.

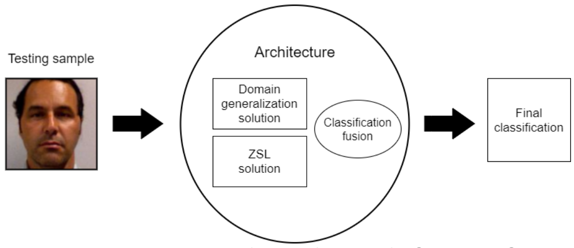

Figure 15: Proposal for the fusion of ZSL with DG

7.2. Future work

Regarding future directions of work, the most crucial task would be experimenting the proposals on a machine capable of providing the best resources possible to run the modifications proposed in this work, and the consecutive comparison with the original results of [29]. Adding to this, since the proposal to modify the triplet loss function took inspiration in [15] and happened to deliver promising results, it would be interesting to compare results with that work. Three additional research directions are briefly discussed in the following.

Fusion strategy with zero-shot learning

The goal of ZSL is to learn from known attack classes to build a model that will be able to also classify samples belonging to previously unseen attack classes, i.e., to classes that were not represented in the training set.

Having a face PAD solution capable of having a good generalization capacity, to different acquisition conditions, and also respond well to unseen attacks, could lead to a robust and trustworthy face PAD approach.

A proposal for combining the two concepts into one solution uses a fusion approach, as illustrated in Figure 15. The proposed architecture applies separately the domain generalization and the ZSL solutions, each trying to solve the task at hand individually, and then combines the achieved results to obtain a final decision.

The biggest challenge in an architecture using two solutions focusing on such different problems would be ”balancing” the scores outputted by both solutions during testing, since there is no obvious way of telling that a unseen class type sample is not fitted for the domain generalization solution and/or a seen sample with adverse conditions is not fitted for the ZSL one.

This topic deserves a lot more research and discussion, not only because domain generalization and ZSL are fairly new concepts that have been recently explored in the face PAD scenario, but also because to the author of this dissertation best knowledge, the possibility of combining these two solutions has not been discussed in PAD literature yet.

Adding another source domain

In [29] the domain generalization was also evaluated considering only two source domains, each one represented by a different database. The results of this experiment were worse when compared to the baseline setting, using three domains, suggesting that having more source domains available, it would possible to learn more generalized cues. This opens the possibility of, with ideal training conditions and suitable hardware, adding another domain to be used during the training stage, and check if more differentiation and generalization cues can effectively be captured.

Designing a rPPG solution

In 3.3 the main issue that prevented the extraction of experimental results, was the incapacity of the algorithm responsible of estimating reliable heart rates from the video samples available in the used datasets. A huge contribution to this failure, was the usage of methods that were developed with health monitoring as the target application, and not PAD, therefore always expecting to find real faces as input and not PAIs.

One possible way to overcome this issue can be by developing a deep learning-based heart rate estimation solution, able to differentiate real and fake samples during training. However, this idea comes with challenges of its own, like for example, finding databases that provide ground truth heart rate measurements to perform training. Also, it is not guaranteed that this solution will have no problems in adapting to typical PAD databases.

References

[1] A. Ali, F. Deravi, and S. Hoque. Spoofing attempt detection using gaze colocation. In *BIOSIG 2013 - Proceedings of the 12th International Conference of the Biometrics Special Interest Group*, 2013.

[2] W. Bao, H. Li, N. Li, and W. Jiang. A liveness detection method for face recognition based on optical flow field. In *Proceedings of 2009 International Conference on Image Analysis and Signal Processing, IASP 2009*, 2009.

[3] G. Boccignone, D. Conte, V. Cuculo, A. D'Amelio, G. Grossi, and R. Lanzarotti. An open framework for remote-PPG methods and their assessment. *IEEE Access*, pages 1–1, 2020.

[4] Z. Boulkenafet, J. Komulainen, X. Feng, and A. Hadid. Scale space texture analysis for face anti-spoofing. In *2016 International Conference on Biometrics, ICB 2016*, 2016.

[5] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face Spoofing Detection Using Colour Texture Analysis. *IEEE Transactions on Information Forensics and Security*, 11(8), 2016.

[6] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid. OULU-NPU: A Mobile Face Presentation Attack Database with Real-World Variations. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 612–618. IEEE, 5 2017.

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

[8] M. Christiaan. Pythonvideopulseratev2.

[9] N. Erdogmus and S. Marcel. Spoofing 2D Face Recognition Systems with 3D Masks. In *BIOSIG 2013 - Proceedings of the 12th International Conference of the Biometrics Special Interest Group*, 2013.

[10] L. Feng, L. M. Po, Y. Li, X. Xu, F. Yuan, T. C. H. Cheung, and K. W. Cheung. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. *Journal of Visual Communication and Image Representation*, 38, 2016.

[11] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11218 LNCS, 2018.

[12] J. Galbally and S. Marcel. Face anti-spoofing based on general image quality assessment. In *Proceedings - International Conference on Pattern Recognition*, 2014.

[13] J. Galbally, S. Marcel, and J. Fierrez. Image quality assessment for fake biometric detection: Application to Iris, fingerprint, and face recognition. *IEEE Transactions on Image Processing*, 23(2), 2014.

[14] Google. Google colaboratory.

[15] Y. Jia, J. Zhang, S. Shan, and X. Chen. Single-Side Domain Generalization for Face Anti-Spoofing. Technical report.

[16] K. Kollreider, H. Fronthaler, and J. Bigun. Evaluating liveness by face images and the structure tensor. In *Proceedings - Fourth IEEE Workshop on Automatic Identification Advanced Technologies, AUTO ID 2005*, volume 2005, 2005.

[17] K. Kollreider, H. Fronthaler, and J. Bigun. Non-intrusive liveness detection by face images. *Image and Vision Computing*, 27(3), 2009.

[18] J. Komulainen, A. Hadid, and M. Pietikäinen. Context based Face Anti-Spoofing. Technical report.

[19] J. Komulainen, A. Hadid, M. Pietikainen, A. Anjos, and S. Marcel. Complementary countermeasures for detecting scenic face spoofing attacks. In *Proceedings - 2013 International Conference on Biometrics, ICB 2013*, 2013.

[20] N. Kose and J. L. Dugelay. Countermeasure for the protection of face recognition systems against mask attacks. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2013*, 2013.

[21] J. Li, Y. Wang, T. Tan, and A. K. Jain. Live face detection based on the analysis of Fourier spectra. In *Biometric Technology for Human Identification*, volume 5404, 2004.

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

[23] W. Liu. Face liveness detection using analysis of Fourier spectra based on hair. In *International Conference on Wavelet Analysis and Pattern Recognition*, volume 2014-January, 2014.

[24] Y. Liu, A. Jourabloo, and X. Liu. Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

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

[26] J. Määttä, A. Hadid, and M. Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *2011 International Joint Conference on Biometrics, IJCB 2011*, 2011.

[27] E. M. Nowara, A. Sabharwal, and A. Veeraraghavan. PPGSecure: Biometric Presentation Attack Detection Using Photopletysmograms. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 56–62. IEEE, 5 2017.

[28] A. Pinto, W. R. Schwartz, H. Pedrini, and A. D. R. Rocha. Using visual rhythms for detecting video-based facial spoof attacks. *IEEE Transactions on Information Forensics and Security*, 10(5), 2015.

[29] R. Shao, X. Lan, J. Li, and P. C. Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, 2019.

[30] D. F. Smith, A. Wiliem, and B. C. Lovell. Face recognition on consumer devices: Reflections on replay attacks. *IEEE Transactions on Information Forensics and Security*, 10(4), 2015.

[31] L. Sun, G. Pan, Z. Wu, and S. Lao. Blinking-based live face detection using conditional random fields. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4642 LNCS, 2007.

[32] Y. Tian and S. Xiang. LBP and Multilayer DCT Based Anti-Spoofing Countermeasure in Face Liveness Detection. *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, 55(3), 2018.

[33] D. Wen, H. Han, and A. K. Jain. Face spoof detection with image distortion analysis. *IEEE Transactions on Information Forensics and Security*, 10(4), 2015.

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