Finger Vein Spoof GANs: Issues in Presentation Attack Detector Training

Andreas Vorderleitner
Dept. of Artificial Intelligence and Human Interfaces,
University of Salzburg (PLUS)
Salzburg, Austria

Andreas Uhl
uhl@cs.sbg.ac.at
Dept. of Artificial Intelligence and Human Interfaces,
University of Salzburg (PLUS)
Salzburg, Austria

Abstract

Four GAN-based I2I translation techniques for unpaired data are employed for the synthesis of biometric finger vein presentation attack instrument (PAI) samples corresponding to three public presentation attack datasets. These synthetic samples are used to train presentation attack detectors (PAD) using distinct feature sets in their classifier. We aim to assess the usefulness of these synthetic data for augmenting PAI datasets, and our analysis reveals that CycleGAN generated PAI samples are best suited to train PAD while DRIT generated data are hardly suited at all. This result corresponds well to visual appearance and quality measures of the synthetic PAI samples. However, it turns out that different types of features used in PAD can lead to very different behaviour of the PAD system trained with synthetic data. For example, Fourier or LBP feature sets must not be used as these respond more to the embedded GAN model fingerprints than to visual similarity of synthetic and real PAI samples. On the other hand, pre-trained neural network features, Haralick features, and surprisingly, also simple features like histograms or localised variance and entropy can be used in the PAD system and lead to stable PAI sample detection results across all datasets and GAN-types (except DRIT) considered. Consequently, results indicate which synthesis technique / feature extraction scheme combinations should be considered when augmenting real PAI samples with synthetic ones in PAD training, and which combinations should be avoided.

CCS Concepts

• **Computing methodologies → Biometrics**; **Reconstruction**; • **Security and privacy → Biometrics**.

Keywords

finger vein recognition, sample synthesis, GAN, presentation attack detection, presentation attack instruments

ACM Reference Format:
Andreas Vorderleitner and Andreas Uhl. 2025. Finger Vein Spoof GANs: Issues in Presentation Attack Detector Training. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31-April 4, 2025, Catania, Italy*. ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3672608.3707714

SAC '25, March 31-April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/10.1145/3672608.3707714

1 Introduction

Presentation attacks (PA) are conducted by either presenting artefacts mimicking real biometric traits (aka "presentation attack instrument" (PAI)) to the biometric sensor to be deceived or by replaying earlier captured biometric sample data on some suited device, thus also attempting to deceive the sensor ("replay attack"). In this work we consider the former attack employing PAIs mounted against finger vein recognition systems. Counter-measures to these types of attacks have of course already been developed and are termed "presentation-attack detection (PAD)" or "anti-spoofing" measures [18]. A comprehensive overview of PAD techniques for vascular data can be found in table 14.1 in [15]. More recent examples are e.g. [23, 24] where a targeted fusion of recognition scheme results is used for PAD and [27] where a customised CNN is trained to detect PAI samples. Also, liveness detection measures can be used against PA, typically by analysing near-infrared videos (in the spectral domain of dorsal hand vein videos [26] or by applying a light vision transformer approach in the Gabor domain of finger vein videos [2]).

The last decade has brought forward several publications that presented multiple ways to potentially fool finger vein-based authentication systems. In first attempts, vascular PAIs are generated as easily as printing a previously captured finger vein sample image on a piece of paper or on overhead projector foil and presenting this printout (eventually manually enhanced) to the sensor (see [21] for a review on these techniques). Current public datasets containing PAI sample data are based on this approach while more advanced techniques involving smartphone displays [21] or modelling finger properties using silicone or beeswax [25] have been developed. Obviously, the generation of PAI samples is tedious work: Generating printouts (manually enhanced) or physical models (in various materials, typically with attached printed vascular structures) and subsequent scanning with a target sensor is required to generate the forged sample data. As a consequence, available PAI sample datasets are of moderate size at best [23] which endangers a statistically relevant assessment of associated security risks. Experimental results in this work confirm this problem: For the most challenging dataset, obtained "BaseLine" results using real PAI samples only are hardly suited for usage in practice (see Experimental Results section), underpinning the need for additional training data. Note that PAI samples are used for two purposes mainly: First, to evaluate the threat posed by such artefacts used in a PA against a particular recognition scheme (vulnerability assessment), and second, to train PAD techniques designed for securing the biometric system [18].

In this work, we focus on the second application case, i.e. training PAD techniques using synthetic data. However, we employ synthetically generated PAI samples created to conduct vulnerability assessment earlier [29]: Based on real PAI sample data of three finger vein spoofing datasets we synthesise PAI samples from given bona fide sample data, which is done by training several different image-to-image (I2I) translation GAN structures. When doing so, the following question arises: Are these synthetic PAI samples fit to be used in both, vulnerability assessment as well as PAD training, i.e. is a dual use of those PAI samples sensible, or do we need to synthesize PAI samples for these two tasks separately ? Besides assessing these synthesised PAI samples in terms of subjective quality and dataset distribution similarity to real PAI sample data (results are taken from previous work [29]), we evaluate the synthetic data in terms of PAD accuracy when used to train a PAD system to discriminate bona fide from real PAI samples, respectively (this is the main objective of this work). Note, that it is **not** the aim of this work to show that we can **improve** PAD system accuracy by using synthetic PAI samples. In fact, this work investigates the effects when replacing existing, *real* PAI samples with the synthetic ones in a PAD system entirely. This setup has been chosen to enable the best-possible and most-accurate comparison setup as the PAI samples used in the comparison (real and synthetic ones) are constructed to even originate from the same bona fide samples. Of course, a realistic PAD system deploying synthetic PAI samples would act differently by augmenting real PAI samples by synthetic ones (instead of replacing them as in our experiments). Our evaluation setup using the synthetic PAI samples only instead of the real ones does not make sense in a real-world system as one could use the real PAI samples right away. In a realistic setting, a model trained to generate synthetic PAI samples would be applied to bona fide samples for which no real PAI samples do exist, to enlarge the training set of the classifier (corresponding evaluations will be done in subsequent work).

The rest of this manuscript is organised as follows: Section 2 describes related work including the selection of suited network architectures for the task at hand. In Section 3 we define the experimental settings with respect to dataset and used evaluation metrics, experimental results are presented in Section 4. The conclusion and outlook to future work is given in Section 5.

2 (Deep-Learning based) Synthesis of PAI Samples

The concept of *Generative Adversarial Networks (GANs [11])* was first introduced by [5] and has become very popular and is also used in a variety of different modified versions. *Image to Image Translation (I2I)* aims to translate an image from a source domain to a target domain. While during this translation the source content should be preserved, the target style should be transferred to the input image I_X. For this I2I task, GANs, in the most imaginable variations, have turned out to be a very good solution [10]. In this work, the source domain are the finger vein images from a specific database and the target domain are the manually created PAI sample images generated by the sensor, which are used to evaluate the presentation attack (PA). Thus, we clearly have an unsupervised 2-domain I2I translation task to solve, in order to supersede the physical construction of presentation attack instruments and their subsequent biometric imaging.

A recent survey on synthetic biometric data [17] reveals, that synthetic generation of vascular data, in particular finger vein samples, has hardly been addressed before. One of the few exceptions is [13], where it was shown that it is indeed possible to generate grey-scale vascular samples (finger vein as well as hand vein data) from corresponding binary features using a learning-based approach (template inversion). An entirely different way for finger vein sample synthesis, using on a model-based approach, has been demonstrated in [9]. Another excellent survey on various aspects of synthetic data (including biometric traits) is presented in [12], confirming the impression that learning-based synthesis of vascular sample data is in its infancy. Still, there are a few further examples employing generative AI techniques: [30] proposed a GAN-based synthesis of a finger vein sample dataset based on the prior generation of a vein pattern image, thus related to both [9] and [13], while [33] applied an end-to-end GAN-based sample generation where the samples are used to augment the training set in deep-learning based finger vein recognition. Similarly, also [31] applies a (Cycle)-GAN-based finger vein sample synthesis approach to improve recognition.

Finally, targeting the synthesis of actual PAI samples, a "Spoof-GAN" has been proposed [7] for fingerprint generation, serving the same purpose as the data generated in this work. A different way to increase the amount of training data for PAD network training is chosen in [19], where usual (non PAI) synthetic fingerprint samples are used for this purpose besides classical PAI samples.

Most similar to this work is [29]. Finger vein PAI samples have been synthesised after training well-known GANs with data from public PAI sample datasets. However, in this paper the synthetic data is evaluated for its appropriateness to conduct a vulnerability assessment, while here, we evaluate the synthetic data for its appropriateness to train a PAD system to discriminate bona fide from real PAI samples, respectively. Note that for vulnerability assessment, synthetic samples need to be suited for impersonation (i.e. synthetic PAI samples are confused with real samples of a person enrolled in a database by the recognition approach used by the biometric system), while for PAD training, synthetic PAI samples "only" need to look / behave similar to real PAI samples according to the employed PAD classifiers' perspective. Results of this study will shed light on the question if PAI samples generated earlier for vulnerability assessment may also be properly employed for PAD training. This would be of advantage of course, as a separate generation of PAD samples for the two different usage scenarios would become obsolete and single datasets with dual usage potential could be made available.

Following [29], the following I2I networks are applied in our task: CycleGAN [34], DistanceGAN [1], DRIT [16], and StarGANv2 [3]. As in [29], we use the network implementations made available by the authors of the original papers, samples are fed into the networks in full size and slightly resized according to the networks need using bicubic (e.g. CycleGAN) or bilinear (e.g. DistanceGAN) interpolation. Augmentations are done within the network as supported, without any additional external augmentation.

Data synthesis is done in a five-fold cross validation, i.e. for each configuration, five different network instances have been trained from scratch to generate their share of the final data. Fold construction prevents to have distinct samples of a single subject in both the training and evaluation sets, respectively (thus we separate subjects in training and evaluation data). The detailed description of which parameters have been used for each network can be retrieved from https://wavelab.at/sources/Vorderleitner23b/ upon publication, based on which the synthetic data can be reproduced.

3 Experimental Settings

3.1 Datasets

The Idiap Research Institute VERA Fingervein Database (VERA) [28] consists of 220 unique fingers captured in 2 sessions from 110 subjects. Each sample has one PAI sample counterpart, which is generated by printing preprocessed samples on high quality paper using a laser printer and enhancing vein contours with a black whiteboard marker afterwards. Images come as full (250×665 pixels) or cropped (150×565 pixels) samples.

The South China University of Technology Finger Vein Database (SCUT) [20] was collected from 6 fingers of 100 subjects captured in 6 acquisition sessions. For PAI sample generation, each finger vein image is printed on two overhead projector films which are aligned and stacked. In order to reduce overexposure, additionally a strong white paper is put in-between the two overhead projector films. Images come as full (640×288 pixels) or cropped samples of variable size.

The Paris Lodron University of Salzburg Finger Vein Spoofing Data Set (PLUS) [25] uses a subset of the PLUS Vein-FV3 dataset as bona fide samples. For PAI sample generation, principle curvature (PC [14]) binarised vein structures from 6 fingers of 22 subjects were printed on paper and sandwiched into a top and bottom made of beeswax. Capturing is done employing two illumination variants (LED and Laser) and using two different levels of vessel thickness. Every sample is of size 192×736 pixels.

In Fig. 1 we display a pair of bona fide and PAI sample images, respectively, from each of the considered datasets. Note that for PLUS data, the two samples look rather differently (so a PAD detector should have an easy job to do), while for VERA and SCUT similarity is much higher, while the PAI samples look much more blurred (and additionally exhibit larger areas of overexposure in case of VERA).

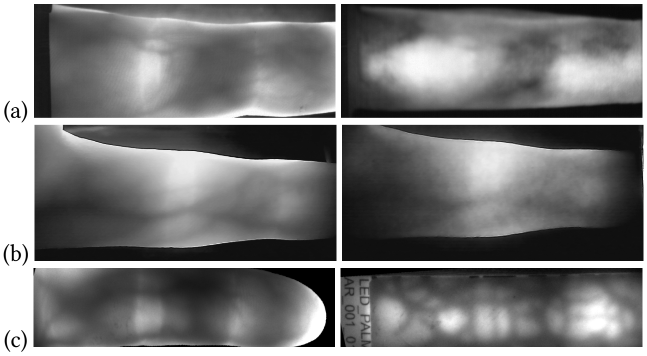

Figure 1: A pair of bona fide and *real* PAI sample images, respectively, from the (a) VERA (b) SCUT, and (c) PLUS datasets.

Note that for each available real PAI sample as contained in the datasets, we have generated the corresponding synthetic PAI sample per generation method (to enable a systematic comparison of real and synthetic samples per instance, respectively).

3.2 Evaluation Methodology

In order to evaluate synthetic visual data, there are several options that can be taken. In any case, we need to keep in mind that the current aim is not to generate "better" data, but to generate data as close as possible to natural (real-world) data. The generation of datasets with custom properties, possibly different from the real data, represents a second stage in the development of such synthetic data.

There is a plethora of techniques to assess visual data, most of the techniques focus on some kind of quality aspect (image quality metrics IQM), and among those, many try to model human perception. This is not exactly our aim as our priority is to generate data which behaves equally to real data when it comes to assess the PA resistance of a finger vein biometric system. Although biometric image quality evaluation algorithms have increasingly been applied in fingerprint, face or iris biometric recognition procedures in recent years, the ISO/IEC 29794:2016 Biometric Sample Quality standard does not yet include a quality evaluation criterion for vein samples [22], while it does for the modalities mentioned before.

As outlined in the Introduction, we aim to assess the effectiveness of synthesised PAI samples in simulating PAD training with these data, as opposed to a simulated PA (vulnerability assessment) as conducted in earlier work on these data [29]. We incorporate existing quality assessment of these synthetic data [29] for the sake of comparison: We include (i) a subjective visual assessment of the correspondence to real-world data (opinion score OS), which is averaged over all samples and uses the same range and interpretation of values as the mean opinion score MOS (five-point, fixed-choice Likert-scale: 1 - good correspondence, 5 - bad correspondence) and (ii) an objective measure of the similarity of visual data considering entire corpora of imagery, the Fréchet Inception Distance (FID) [8]. For both metrics, smaller values indicated better correspondence to the real PAI sample data.

For the performance assessment in PAD training, we train kNN (k-nearest neighbour) classifiers to facilitate straightforward applicability for all feature sets considered (see below, except for FscratchNN). Note that kNN is chosen to concentrate the attention to the effects of the different feature extraction schemes, which is done most effectively using the kNN as it does not really learn but only memorizes the training-set and selects classification results based on feature vector proximity. For "BaseLine" results, we train the classifier to discriminate between bona fide and *real* PAI samples as provided by the datasets. For assessing the synthetic data, we train the classifier to discriminate between bona fide and synthetic PAI samples, respectively. As the aim is to conduct "real" PAD, in both cases we test the ability of the resulting classifier to discriminate between bona fide and *real* PAI samples. Thus, while the testing is done in the same manner (considering real-life application), the training is different according to what we mean to determine. As the available datasets are rather limited in size, we apply the leave-one-out cross validation protocol (for the non-learning-based features) and a five-fold cross validation for the learning based ones.

To evaluate the effectiveness of the proposed PAD approach, results are reported in compliance with ISO/IEC 30107-3:2017. Since a presentation attack detection mechanism is a binary classifier, four outcomes are possible: correctly classified as attack (true positives TP), wrongly classified as attack (false positive FP), correctly classified as bona fide (true negative TN) and wrongly classified as bona fide (false negative FN). According to the standard, we report Attack Presentation Classification Error Rate (APCER - proportion of attack presentations incorrectly classified as bona fide presentations), Bona Fide Presentation Classification Error Rate (BPCER - proportion of bona fide presentations incorrectly classified as presentation attacks), and Average Classification Error Rate (ACER = (APCER + BPCER) / 2) as follows:

$$APCER = \frac{FN}{FN + TP} \quad , \quad BPCER = \frac{FP}{FP + TN} \; .$$

The features employed as described in the following subsections have been selected as these have often been used to characterize textured content. As the texture of tissue and vessels seems to vary significantly when comparing bona fide and PAI samples, respectively, texture-oriented descriptors constitute a natural choice for the task at hand. In addition, some generic visual descriptors have been selected.

3.2.1 Classical Features. Grey-value *histograms* are formed using 10 equally sized bins, k in the kNN is set to 3 and Euclidean distance is used to compare histograms. For the *variance* (*entropy*) feature-vector, the samples are cut into 40×40 (50×50) pixels blocks for each of which the variance (entropy) is computed (k is set to 3 (5), Euclidean distance is used to compare feature-vectors). For the *fractal dimension* feature vector, the samples are binarised with 9 uniformly distributed thresholds across the grey-value range and box-counting fractal dimension is computed resulting in 9-component vectors which are compared by Euclidean distance (k in kNN is set to 3 (SCUT), 5 (VERA), and 7 (PLUS)). For the *Haralick* feature vector, the underlying grey-level co-occurrence matrix (GLCM) considers not a single direction but all directions combined (i.e. number of neighbouring pixels per grey-value combination) at a certain distance. k is set to 9, while GLCM distance d as well as Haralick feature H used are found in parameter optimisation: VERA: d=2, H=3, SCUT: d=3, H=3 & 12, PLUS: d=1, H=9. Feature vectors are compared by Euclidean distance. The Local Binary Pattern (*LBP*) feature vector is obtained by using uniform LBP with radii 1 and 2 (two histograms concatenated which are compared using the Wasserstein distance) and setting k in the kNN to 7. Binary Statistical Image Features (*BSIF*) uses filters learned by ICA, however, we have used filters pre-trained on iris data[1]. Responses of the 10 11×11 pixel filters are mean-binarised, flattened and Hamming distance is used for comparison. k is set to 5, some configurations of dataset and GAN-type provide slightly better results for other k-values. The *Fourier* feature vector consists of the energy of Fourier coefficients computed from 30 equally-spaced band pass filters, again compared by Euclidean distance. Unless noted otherwise, all 30 bands are used, and best results are observed for k between 10 and 25 (variations are small for k larger than 5 though).

3.2.2 Learning-based Features. *DenseSIFT* and *DenseHOG* features vectors are obtained by a Bag-of-Visual-Words (BOW) approach applied to a set of keypoints computed on centers of 3×3 pixel blocks. The keypoint descriptors computed on this regular ("dense") grid are k-means clustered and the clusters form the bin-centers of the histograms describing any new image. The k in the kNN classifier is optimised for each dataset and GAN, k in k-means clustering has been found to perform well at k=9 (SIFT) and K=11 (HOG), resulting in 9-bin and 13-bin histograms which are compared by Euclidean distance. *TransferNN* denotes using the EfficientNet-B0 in transfer-learning mode, i.e. it is used as a pre-trained (on ImageNet data) feature extractor, from which the activations of the last layer are fed into the kNN, using k=5 and comparing vectors using Euclidean distance. Finally, *FscratchNN* denotes a small custom network, trained from scratch (i.e. random weight initialisation) on our data. The architecture is extremely shallow, consisting of a single convolutional layer with ReLU activation and subsequent MaxPooling layer. Next, a flatten layer prepares the data to be input into a Dense layer again with ReLU activation before resorting to the Output layer (with a sigmoid activation function). Adam optimisation is employed, the loss function used is BinaryCrossentropy.

4 Experimental Results

For each of the three datasets and the four GAN-types, we present a visual example of a synthetic PAI sample for qualitative analysis in Fig. 2. When comparing to Fig. 1, we observe that the PAI samples from the SCUT dataset are difficult to synthesise properly, as except for the CycleGAN result, the samples lack in clear vascular structure. The DRIT data look rather disappointing overall, as even the PLUS data clearly lack in detail (which is rather prominent and much better generated by the other GAN-types).

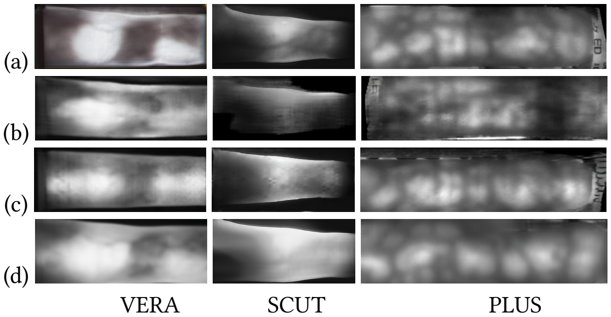

Figure 2: Example synthetic PAI sample images: (a) CycleGAN (b) DistanceGAN (c) StarGANv2 (d) DRIT

In Table 1 we present the quantitative results of our assessment for data generated from the VERA dataset. In terms of FID and OS, the numerical values clearly favour DistanceGAN over DRIT, thus confirming the visual impression. StarGANv2 shows even better values in that respect, while CycleGAN surprisingly exhibits the worst FID value and OS close to DRIT, contradicting the visual quality. In the subsequent lines of Table 1, we report results of

[1] https://github.com/CVRL/domain-specific-BSIF-for-iris-recognition

PAD detector performance, considering BaseLine results as well as results obtained using synthetic PAI samples in PAD training. Error rates have been rounded to integer in all subsequent result tables as (i) we present a significant number of numerical results and decimal figures might cloud the major message conveyed, and (ii) for several feature types we already present an error-range due to different parameter settings, such that too much detail in single number results would not make much sense. Note that in increasing darkness of gray levels, we emphasise decreasing PAD accuracy - i.e., the darker, the worse (medium gray for ACER between 21% and 40% and dark grey above 40%, respectively). Also note that for the SCUT and PLUS datasets, experiments are conducted on multiple instances of synthetic data differing by minor parameter variations in the synthesis process (see https://wavelab.at/sources/Vorderleitner23b/ for details), also FscratchNN experiments are repeated with varying random initialisations for all datasets. If these experiments cause variations in classification accuracy, a range is given instead of a single value. In section 4.1 a single result from this range is further detailed for illustrative purposes.

Table 1: VERA - Quantitative results: FID, OS, and ACER (BaseLine and training with four variants of GAN synthesised PAI samples).

	BaseLine	**CycleGAN**		**DistanceGAN**		**StarGANv2**		**DRIT**	
		FID	**OS**	**FID**	**OS**	**FID**	**OS**	**FID**	**OS**
		189	3.0	28	2.5	24	2.0	106	3.5
Method	**ACER** [%]	**ACER** [%]		**ACER** [%]		**ACER** [%]		**ACER** [%]	
Histogram	1	4		2		2		12	
Variance	3	4		2		2		3	
Entropy	1	1		0		1		2	
FactDim	2	35		25		30		40	
Haralick	0	8		0		1		0	
LBP	4	29		50		50		50	
BSIF	0	28		32		1		4	
Fourier	7	50		50		50		50	
DenseSIFT	1	3		2		2		13	
DenseHOG	0	5		3		2		17	
TransferNN	1	3		3		3		3	
FscratchNN	2	46-53		68-62		24-27		60-64	

First of all, it is evident that all used feature sets can be used to train a PAD system using real data, as the "BaseLine" column exhibits fairly low ACER values. Furthermore, PAD training using the synthetic data set is feasible in principle (excellent results for all GAN-types using e.g. Entropy, Variance, Haralick, and TransferNN), however, it is also evident that the success of this approach is dependent on the feature set employed (random-guessing like results for all GAN-types using LBP, Fourier, and FscratchNN). Also, not all GAN-types are equally useful, in particular we see DRIT with the worst results while the other three are about en par, with StarGAN in the lead. So it seems, that visual appearance is more reliable for predicting PAD training performance than quantitative measures of quality.

Subsequently, we discuss quantitative results obtained on the SCUT dataset which are shown in Table 2. Comparing the different GAN-types in terms of FID and OS, the visual impression is numerically confirmed. Only CycleGAN exhibits low values (indicating good quality), whereas the other three GANs lead to high values (only SarGANv2 has a somewhat lower FID value than the other two). The overall visual impression of this table is different compared to the VERA results with many more grey fields appearing (i.e. PAD accuracy is lower in general). Also BaseLine accuracy is at a less useful level for this more challenging dataset. We have two features with excellent results when the PAD system is trained on synthetic data (for all GAN-types: Haralick and TransferNN), but many more with very poor performance making them useless (i.e. again LBP, Fourier, and FscratchNN, but also BSIF, DenseSIFT and DenseHOG). Again, the DRIT generated data is performing worst, while CycleGAN generated data gives the best results (confirming visual impression and quantitative quality measures).

Table 2: SCUT - Quantitative results: FID, OS, and ACER (BaseLine and training with four variants of GAN synthesised PAI samples). Lines marked by "*" will be analysed in section 4.1.

	BaseLine	**CycleGAN**		**DistanceGAN**		**StarGANv2**		**DRIT**	
		FID	**OS**	**FID**	**OS**	**FID**	**OS**	**FID**	**OS**
		41	2.0	110	4.5	74	4.0	122	4.0
Method	**ACER** [%]	**ACER** [%]		**ACER** [%]		**ACER** [%]		**ACER** [%]	
Histogram*	2	2-4		3-6		2		38-44	
Variance*	9	8		8		4-6		20	
Entropy*	4	6-11		36-37		2		46	
FactDim	6	15-18		29-33		13-14		25-29	
Haralick	0	0		0		0		0	
LBP*	4	30-33		50		50		50	
BSIF	0	40-50		38-41		50		45-47	
Fourier*	18	50		50		50		50	
DenseSIFT	6	35-37		44-48		16-18		43-44	
DenseHOG	6	31-34		47-48		21-29		47-49	
TransferNN	1	3		3		3		3	
FscratchNN*	3	73-76		60-64		47-53		48-52	

Finally, we discuss results computed on the PLUS dataset. When comparing the qualitative visual impression of the generated samples to the quantitative findings in Table 3, we observe a similar ranking (at least in terms of FID): CycleGAN is in the lead, DistanceGAN and StarGANv2 about the same, and DRIT worst. CycleGAN is in the lead also in terms of OS, while DRIT takes the second place in this category (eventually, the displayed PAI sample is of untypically poor quality for DRIT).

Table 3: PLUS - Quantitative results: FID, OS, and ACER (BaseLine and training with four variants of GAN synthesised PAI samples). Lines marked by "*" will be analysed in section 4.1.

	BaseLine	**CycleGAN**		**DistanceGAN**		**StarGANv2**		**DRIT**	
		FID	**OS**	**FID**	**OS**	**FID**	**OS**	**FID**	**OS**
		57	2.0	111	4.5	128	4.5	163	3.5
Method	**ACER** [%]	**ACER** [%]		**ACER** [%]		**ACER** [%]		**ACER** [%]	
Histogram	0	0		0		1		0	
Variance	1	6-9		1-3		2		13-16	
Entropy	0	0		0		0		0	
FactDim	1	1		2		3-5		59-61	
Haralick	0	0		0		0-3		0-21	
LBP*	0	1		50		50		50	
BSIF	0	3-4		2		1-2		2-3	
Fourier*	4	50		50		50		50	
DenseSIFT	1	2		2		2		2	
DenseHOG	0	0-2		0-2		0-2		11-44	
TransferNN	1	3		3		3		3	
FscratchNN*	3	69-75		54-65		45-50		54-65	

For the PLUS dataset, we see the overall best PAD accuracy results among the three datasets considered. Excellent BaseLine performance, and also excellent accuracy values for PAD training with synthetic data except for LBP, Fourier, and FscratchNN features. This is true for all GAN-types but DRIT, for which we observe three additional feature types with degraded accuracy (Variance, FractDim, DenseHOG). Obviously, the quite distinct structure of the (real and synthetic) PAI samples as compared to bona fide samples helps to result in good PAD accuracy.

The results presented show significant variation in error rates depending on the combination of feature extraction methods, data generation techniques, and dataset considered. For the same feature, some synthetic data generation methods produce very low error rates, while others produce extremely high ones. In a vast majority of cases, the purely quality-oriented metrics (FID and OS) provide a reasonable prediction for the outcome in terms of PAD accuracy. Consequently, DRIT and DistanceGAN generated PAI samples cannot be recommended in the context of PAD training. The following subsection aims to shed light on the reason for the poor performance of some features used.

4.1 Results Discussion

In this section we aim to have a closer look at the results, in particular to reveal the reason for particularly poor performing settings. We facilitate this by analysing the per-class PAD classification accuracies, i.e. BPCER and APCER, respectively. Table 4 shows the SCUT dataset results for a selection of poorly performing feature sets. We are confronted with an interesting result – for all but two cases (out of overall 24, i.e. four GAN types times six feature sets), BPCER is significantly lower than APCER. The Fourier feature set takes this to the extreme: While all bona fide samples are correctly classified (BPCER = 0%), none of the PAI samples is, resulting in APCER = 100% (we see the identical behaviour for LBP except for the CycleGAN data).

Table 4: SCUT - detailed quantitative results: BPCER and APCER for each GAN.

	CycleGAN		DistanceGAN		StarGANv2		DRIT	
	BPCER	APCER	BPCER	APCER	BPCER	APCER	BPCER	APCER
Method	[%]		[%]		[%]		[%]	
Histogram	3	3	3	4	3	1	3	84
Variance	8	7	3	13	7	5	2	37
Entropy	1	22	0	71	1	4	0	92
LBP	2	44	0	100	0	100	0	100
Fourier	0	100	0	100	0	100	0	100
FscratchNN	35	70	9	92	66	39	1	99

Table 5 shows corresponding results of poorly performing feature sets for the PLUS dataset. We observe exactly the same behaviour in that for those feature sets, it is the complete inability of the classifier to correctly identify PAI samples which causes the poor results (for LBP and Fourier features APCER = 100% consistently). The FscratchNN results exhibit a more balanced poor performance, only the DRIT results also point into the same direction as seen for the other feature sets.

Table 5: PLUS - detailed quantitative results: BPCER and APCER for each GAN.

	CycleGAN		DistanceGAN		StarGANv2		DRIT	
	BPCER	APCER	BPCER	ACPER	BPCER	APCER	BPCER	APCER
Method	[%]		[%]		[%]		[%]	
LBP	0	1	0	100	0	100	0	100
Fourier	0	100	0	100	0	100	0	100
FscratchNN	52	50	50	46	48	50	25	72

What is happening here ? These results are associated with the ambiguity which data properties are emphasised by the used feature extraction. In the setup of our PAD training using synthetic PAI samples, a feature extraction scheme may focus on two different things: First, the intended discrimination into bona fide vs. PAI samples (which is based on the visual similarity of real and synthetic PAI samples). Second, a feature extraction scheme may accentuate properties that are used to discriminate real from synthetic images [6], as the bona fide samples are real natural pictorial data, while the GAN-generated PAI samples are synthetic. If the second case applies, the PAD system will recognise bona fide samples as well as real PAI samples as being real image data, and will classify both into the "real image" category, while the second category ("synthetic data", or the intended PAI sample data) stays empty. This is 100% correct for bona fide images, but 0% correct for the PAI sample data. When looking into the results shown in tables 4 and 5, we see this effect occurring for almost all feature sets with poor overall PAD accuracy. Most notably, this happens with full manifestation for the Fourier features.

Why is this happening with such peculiarity for the Fourier feature set ? It is well known that synthetic image data generated by GANs carries a (more or less) imperceptible "model fingerprint" which can be even used to identify the source GAN. While these fingerprints can be well detected in the auto-correlation domain, the most prominent domain for detection and visualisation is the Fourier domain [4, 32]. Thus, of course bona fide samples as well as real PAI samples do not carry such a fingerprint, while synthetic PAI samples do. Figure 3 illustrates this property.

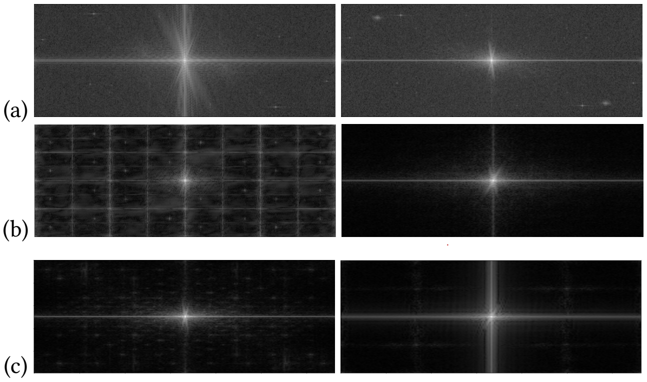

Figure 3: FFT magnitude of VERA dataset (a) bona fide and PAI samples (b) CycleGAN and DistanceGAN synthetic PAI samples (c) StarGan and DRIT synthetic PAI samples.

The most obvious difference between real data (line 1) and synthetic data (lines 2 & 3) in the figure is that the Fourier magnitude of real data looks brighter overall. This is not a normalisation error, but the rather low quality real data have dispersed their energy across almost all frequency bands (resulting in many non-zero coefficients which generate the overall brighter appearance), while this is not the case for the synthetic data exhibiting many more close-to-zero coefficients, appearing darker. Additionally, upon closer inspection, it gets evident that the actual model fingerprints are also present indicated by periodic patterns, best seen in the CycleGAN data, but also present in the data generated by the other three GAN-types. Thus it is entirely clear that Fourier features discriminate real from synthetic data but not bona fide from PAI samples, respectively, as the Fourier domain mainly emphasizes differences between real and synthetic image data.

On the other hand, the stable performance of the TransferNN feature set across all data sets and GAN-types may be attributed to the fact that the underlying network has been trained on visually different classes from the ImageNet data, which differs by specific visual (perceptually relevant) clues but certainly not by specific (nearly imperceptible) features in the Fourier domain. For most other feature types, they also lean more towards taking advantage of visual similarity, however, interestingly, only LBP sets the focus more towards the model fingerprints as well. The small network trained from scratch is the only feature set which seems to be kind-of undecided which content it is more concerned with.

5 Conclusion

Among the four I2I translation GANs we have identified a clear ranking: CycleGAN generated PAI samples are best suited to train PAD (note the correspondence to [29], where CycleGAN synthesised PAI samples are found to be best suited for vulnerability assessment). DRIT generated data are hardly suited at all in the context of PAD. DistanceGAN and StarGANv2 rank in-between. Overall, this result corresponds well to visual appearance and objective quality measures of the synthetic PAI samples. Thus, we may conclude that CycleGAN generated finger vein PAI samples enable dual usage, i.e. assessing system vulnerability against PA and training PAD techniques.

However, there are some particularities connected to the nature of the PAD training: Different types of features used in PAD can lead to very different behaviour of the PAD system trained with synthetic data. For example, Fourier or LBP feature sets must not be used as these respond more to the embedded GAN model fingerprints than to visual similarity of synthetic and real PAI samples. On the other hand, pre-trained neural network features, Haralick features, and surprisingly, also simple features like histograms or localised variance and entropy can be used in the PAD system and lead to stable PAI sample detection results across all datasets and GAN-types (except DRIT) considered.

Thus, to apply the findings in this paper for designing a finger vein PAD system employing synthetic PAI sample data the following steps must be taken:

(1) For the biometric system to be protected, (real) PAI samples need to be available in addition to the bona fide sample data.
(2) Generate synthetic PAI samples using CycleGAN using a model trained on the available bona fide and real PAI samples, respectively. The synthetic PAI samples are derived from bona fide samples for which no real PAI samples do exist. Results in [29] demonstrate that these PAI samples can be used in vulnerability assessment as well.
(3) Select a feature representation identified to be applicable in this work and train the PAD system with bona fide sample data (class 1) vs. real and synthetic PAI samples (class 2).

Future work will investigate the actual application of the synthetic data found to be suited (in this work) to enrich the training data for a two-class PAD system. For this purpose, both real as well as synthetic PAI data derived from distinct users shall be used for training the attack class. The PAD system will be based on a suited feature representation (as identified in this work). Additionally, we will apply diffusion models and non-I2I transform GAN types for generating PAI samples "from scratch", i.e. only suited for PAD training but not for impersonation, and will compare the results of correspondingly trained PADs to this work. We will also investigate the impact of recent GAN model fingerprint removal techniques on the obtained results.

Acknowledgments

This research was partially funded by the Austrian Science Fund FWF under the project no. I4272. We acknowledge the significant contributions of the students participating in the Image Processing and Imaging Lab (Winter Term 2023/2024) at the University of Salzburg.

References

[1] Sagie Benaim and Lior Wolf. 2017. One-Sided Unsupervised Domain Mapping. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)* (Long Beach, California, USA). Curran Associates Inc., Red Hook, NY, USA, 752–762.

[2] Liukui Chen, Tengwen Guo, Li Li, Haiyang Jiang, Wenfu Luo, and Zuojin Li. 2023. A Finger Vein Liveness Detection System Based on Multi-Scale Spatial-Temporal Map and Light-ViT Model. *Sensors* 23, 24 (2023). https://doi.org/10.3390/s23249637

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR'20*. 8188–8197. https://doi.org/10.1109/cvpr42600.2020.00821

[4] Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. 2023. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 973–982. https://doi.org/10.1109/CVPRW59228.2023.00104

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative Adversarial Networks. *Commun. ACM* 63, 11 (oct 2020), 139–144. https://doi.org/10.1145/3422622

[6] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva. 2021. Are GAN Generated Images Easy to Detect? A Critical Analysis of the State-Of-The-Art. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6. https://doi.org/10.1109/ICME51207.2021.9428429

[7] Steven A. Grosz and Anil K. Jain. 2023. SpoofGAN: Synthetic Fingerprint Spoof Images. *IEEE Trans. on Information Forensics and Security* 18 (2023), 730–743. https://doi.org/10.1109/TIFS.2022.3227762

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)* (Long Beach, California, USA). Curran Associates Inc., Red Hook, NY, USA, 6629–6640.

[9] Fieke Hillerström, Ajay Kumar, and Raymond Veldhuis. 2014. Generating and Analyzing Synthetic Finger Vein Images. In *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG'14)*. 121–132.

[10] Henri Hoyez, Cédric Schockaert, Jason Rambach, Bruno Mirbach, and Didier Stricker. 2022. Unsupervised Image-to-Image Translation: A Review. *Sensors* 22, 21 (2022). https://doi.org/10.3390/s22218540

[11] Abdul Jabbar, Xi Li, and Bourahla Omar. 2021. A Survey on Generative Adversarial Networks: Variants, Applications, and Training. *ACM Comput. Surv.* 54, 8, Article 157 (2021), 49 pages. https://doi.org/10.1145/3463475

[12] Indu Joshi, Marcel Grimmer, Christian Rathgeb, Christoph Busch, Francois Bremond, and Antitza Dantcheva. 2022. Synthetic Data in Human Analysis: A Survey. https://doi.org/10.48550/ARXIV.2208.09191

[13] Christof Kauba, Simon Kirchgasser, Vahid Mirjalili, Andreas Uhl, and Arun Ross. 2021. Inverse Biometrics: Generating Vascular Images from Binary Templates. *IEEE Transactions on Biometrics, Behavior, and Identity Science* 3, 4 (2021), 464–478. https://doi.org/10.1109/TBIOM.2021.3073666

[14] Christof Kauba and Andreas Uhl. 2019. An Available Open-Source Vein Recognition Framework. In *Handbook of Vascular Biometrics*, Andreas Uhl, Christoph

Busch, Sebastien Marcel, and Raymond Veldhuis (Eds.). Springer Nature Switzerland AG, Cham, Switzerland, Chapter 4, 113–142. https://doi.org/10.1007/978-3-030-27731-4_4

[15] Jascha Kolberg, Marta Gomez-Barrero, Sushma Venkatesh, Raghavendra Ramachandra, and Christoph Busch. 2020. *Presentation Attack Detection for Finger Recognition.* Springer International Publishing, Cham, 435–463. https://doi.org/10.1007/978-3-030-27731-4_14

[16] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2018. Diverse Image-to-Image Translation via Disentangled Representations. In *Proceedings of the European Conference on Computer Vision ECCV'18.* 36–52. https://doi.org/10.1007/978-3-030-01246-5_3

[17] Andrey Makrushin, Andreas Uhl, and Jana Dittmann. 2023. A Survey On Synthetic Biometrics: Fingerprint, Face, Iris and Vascular Patterns. *IEEE ACCESS* 11 (2023), 33887–33899. https://doi.org/10.1109/ACCESS.2023.3250852

[18] S. Marcel, J. Fierrez, and N. Evans (Eds.). 2023. *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection and Vulnerability Assessment.* Springer. https://doi.org/10.1007/978-981-19-5288-3

[19] Sandip Purnapatra et al. 2023. Presentation Attack Detection with Advanced CNN Models for Noncontact-based Fingerprint Systems. In *Proceedings of the 11th International Workshop on Biometrics and Forensics (IWBF'23)* (April 20 - April 21). Barcelona, Spain, 1–6.

[20] X. Qiu, W. Kang, S. Tian, W. Jia, and Z. Huang. 2018. Finger Vein Presentation Attack Detection Using Total Variation Decomposition. *IEEE Transactions on Information Forensics and Security* 13, 2 (2018), 465–477.

[21] R. Raghavendra and C. Busch. 2015. Presentation Attack Detection Algorithms for Finger Vein Biometrics: A Comprehensive Study. In *11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS'15).* 628–632.

[22] Oliver Remy, Jutta Hämmerle-Uhl, and Andreas Uhl. 2022. Fingervein Sample Image Quality Assessment using Natural Scene Statistics. In *2022 International Conference of the Biometrics Special Interest Group (BIOSIG'22).* 1–6. https://doi.org/10.1109/BIOSIG55365.2022.9896974

[23] Johannes Schuiki, Michael Linortner, Georg Wimmer, and Andreas Uhl. 2022. Attack Detection for Finger and Palm Vein Biometrics by Fusion of Multiple Recognition Algorithms. *IEEE Transactions on Biometrics, Behavior, and Identity Science* 4, 4 (2022), 544 – 555. https://doi.org/10.1109/TBIOM.2022.3212836

[24] Johannes Schuiki, Michael Linortner, Georg Wimmer, and Andreas Uhl. 2023. Extensive Threat Analysis of Vein Attack Databases and Attack Detection by Fusion of Comparison Scores. In *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection and Vulnerability Assessment*, Sebastien Marcel, Julian Fierrez, and Nicholas Evans (Eds.). Springer Nature Singapore, Singapore, 467–487. https://doi.org/10.1007/978-981-19-5288-3_17

[25] Johannes Schuiki, Bernhard Prommegger, and Andreas Uhl. 2021. Confronting a Variety of Finger Vein Recognition Algorithms With Wax Presentation Attack Artefacts. In *Proceedings of the 9th IEEE International Workshop on Biometrics and Forensics (IWBF'21)* (May 6 - May 7). Rome, Italy, 1–6.

[26] Johannes Schuiki and Andreas Uhl. 2020. Improved Liveness Detection in Dorsal Hand Vein Videos using Photoplethysmography. In *Proceedings of the IEEE 19th International Conference of the Biometrics Special Interest Group (BIOSIG 2020)* (September 16 - September 18). Darmstadt, Germany, 57–65.

[27] Kashif Shaheed, Aihua Mao, Imran Qureshi, Qaisar Abbas, Munish Kumar, and Xingming Zhang. 2022. Finger-Vein Presentation Attack Detection using Depthwise Separable Convolution Neural Network. *Expert Systems with Applications* 198 (03 2022), 116786. https://doi.org/10.1016/j.eswa.2022.116786

[28] P. Tome, R. Raghavendra, C. Busch, S. Tirunagari, N. Poh, B. H. Shekar, D. Gragnaniello, C. Sansone, L. Verdoliva, and S. Marcel. 2015. The 1st Competition on Counter Measures to Finger Vein Spoofing Attacks. In *International Conference on Biometrics (ICB'15).* 513–518.

[29] Andreas Vorderleitner, Jutta Hämmerle-Uhl, and Andreas Uhl. 2023. Finger Vein Spoof GANs: Can we Supersede the Production of Presentation Attack Artefacts?. In *22th International Workshop on Digital Forensics and Watermarking (IWDW'23)* (November 25 - November 26) *(Springer LNCS, Vol. 14511).* Jinan, China, 109–124. https://doi.org/10.1007/978-981-97-2585-4

[30] Hanwen Yang, Peiyu Fang, and Zhiang Hao. 2021. A GAN-Based Method for Generating Finger Vein Dataset. In *Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence* (Sanya, China) *(ACAI '20).* Association for Computing Machinery, New York, NY, USA, Article 18, 6 pages. https://doi.org/10.1145/3446132.3446150

[31] Wenming Yang, Changqing Hui, Zhiquan Chen, Jing-Hao Xue, and Qingmin Liao. 2019. FV-GAN: Finger Vein Representation Using Generative Adversarial Networks. *IEEE Transactions on Information Forensics and Security* 14, 9 (2019), 2512–2524. https://doi.org/10.1109/TIFS.2019.2902819

[32] N. Yu, L. Davis, and M. Fritz. 2019. Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE Computer Society, Los Alamitos, CA, USA, 7555–7565. https://doi.org/10.1109/ICCV.2019.00765

[33] Jianfeng Zhang, Zhiying Lu, Min Li, and Haopeng Wu. 2019. GAN-Based Image Augmentation for Finger-Vein Biometric Recognition. *IEEE Access* 7 (2019), 183118–183132. https://doi.org/10.1109/ACCESS.2019.2960411

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV).* 2242–2251. https://doi.org/10.1109/ICCV.2017.244 ISSN: 2380-7504.