SpoofGAN: Synthetic Fingerprint Spoof Images

Steven A. Grosz and Anil K. Jain, Life Fellow, IEEE

Abstract—A major limitation to advances in fingerprint presentation attack detection (PAD) is the lack of publicly available, large-scale datasets, a problem which has been compounded by increased concerns surrounding privacy and security of biometric data. Furthermore, most state-of-the-art PAD algorithms rely on deep networks which perform best in the presence of a large amount of training data. This work aims to demonstrate the utility of synthetic (both bona fide and PA style) fingerprints in supplying these algorithms with sufficient data to improve the performance of fingerprint PAD algorithms beyond the capabilities when training on a limited amount of publicly available “real” datasets. First, we provide details of our approach in modifying a state-of-the-art generative architecture to synthesize high quality bona fide and PA fingerprints. Then, we provide quantitative and qualitative analysis to verify the quality of our synthetic fingerprints in mimicking the distribution of real data samples. We showcase the utility of our synthetic bona fide and PA fingerprints in training a deep network for fingerprint PAD, which dramatically boosts the performance across three different evaluation datasets compared to an identical model trained on real data alone. Finally, we demonstrate that only 25% of the original (real) dataset is required to obtain similar detection performance when augmenting the training dataset with synthetic data. We make our synthetic dataset and model publicly available to encourage further research on this topic: https://github.com/groszste/SpoofGAN.

Index Terms—Generative adversarial networks, fingerprint presentation attack detection, fingerprint spoof detection, synthetic fingerprint generation.

I. INTRODUCTION

FINGERPRINT recognition has had a long history in person identification due to the purported uniqueness and permanence of fingerprints, originally pointed out by Sir Francis Galton in his 1892 book titled “Finger Prints” [1] and reaffirmed in many works over the last century; including the well-known studies on the individuality and longitudinal permanence of fingerprint recognition [2], [3]. Clearly, a significant contributor to their wide spread adoption is the high level of verification performance achieved by state-of-the-

art (SOTA) algorithms for automated fingerprint recognition.¹

However, despite the impressive accuracy achieved to date by the top-performing fingerprint recognition algorithms, there remain many on-going efforts to further improve the capabilities of fingerprint recognition systems - especially in terms of recognition speed and system security. As a result, there has been a recent push toward deep neural network (DNN) based models for fingerprint recognition [5], [6], [7], [8], [9], [10], [11]. These compact, fixed-length embeddings can be matched efficiently and combined with homomorphic encryption for added security [12]. For a more exhaustive account of existing deep learning approaches to fingerprint recognition and other biometric modalities, interested readers are encouraged to consult one of the many surveys on deep learning in biometrics (e.g., [13], [14]).

Indeed, this push toward DNN-based fingerprint recognition comes in the wake of the success demonstrated in the

¹Today's top-performing algorithm in the FVCongoing 1:1 hard benchmark achieved a False Non-Match Rate (FNMR) of just 0.626% at a False Match Rate (FMR) of 0.01% [4].
face recognition domain in applying DNN models to face recognition, which was aided by the availability of large-scale face recognition databases which were easily crawled from the web; despite the many ethical and privacy concerns which have led to many of these datasets to be recalled today. Arguably, at least in part, the reason for the delayed adoption of DNNs for fingerprint recognition has been the lack of publicly available, large-scale fingerprint recognition datasets and increased scrutiny over privacy of biometric data, which has led to many works to generate synthetic fingerprint images [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29].

Similarly, there has been an increased interest in DNN-based models for fingerprint presentation attack detection (PAD); i.e., spoof detection, where the scale and amount of publicly available data is also limited. Table I gives a list of the publicly available fingerprint PA datasets. Compared to the largest, public fingerprint recognition dataset, NIST Special Database 302 [30], which contains fingerprints from 2,000 unique fingers, the largest publicly available fingerprint PA datasets, e.g., the LivDet competition datasets, contain at most 1,000 unique fingers (for the Swipe sensor in LivDet 2013 [31]). Compounding the problem is the difficulty in collecting large-scale fingerprint PA datasets due to the increased time and complexity in fabricating and imaging artefacts mimicking realistic fingerprint ridge-valley structures. All of which motivates the potential of synthetic data as a viable alternative; however, to the best of our knowledge, there does not exist a synthetic fingerprint PA generator to fill the gap between the amount of publicly available fingerprint PA data and training of data-hungry deep learning-based models.

To address the lack of large-scale fingerprint PA datasets, we propose SpoofGAN. Inspired by the impressive results of the recently proposed PrintsGAN [15], SpoofGAN is a multi-stage generative architecture to fingerprint generation. SpoofGAN is different from PrintsGAN in the following ways:

- Generation of plain print fingerprints, which compared to the rolled prints generated by PrintsGAN, are more representative of the publicly available PA fingerprint datasets and exhibit different textural characteristics, distortions, etc.
- Ability to synthesize representations of both bona fide (i.e., live) and PA fingerprint images of the same finger.
- Replacing the learned warping and cropping module with a statistical, controllable non-linear deformation model to synthesize multiple, realistic impressions per finger. This allows us to control the degree of distortion applied.

In this work, we make the distinction that real fingerprint images are those captured on a fingerprint reader by either a real human finger or physical artefact (presentation attack) which mimics a fingerprint ridge structure, whereas synthetic fingerprint images are digital renderings of fingerprint images. However, there can be both real bona fide and real PA fingerprint images, as well as synthetic representations of bona fide and synthetic representations of PA fingerprint images. For clarification, we use the following four classifications in this work:

- **Real bona fide** fingerprint image: fingerprint images captured from a real human finger on a fingerprint reader.
- **Synthetic bona fide** fingerprint image: fingerprint images generated by a deep learning model trained on real fingerprint images.
- **Real presentation attack** fingerprint image: fingerprint images captured from a real human finger on a fingerprint reader, which has been fabricated into a presentation attack artefact.
- **Synthetic presentation attack** fingerprint image: fingerprint images generated by a deep learning model trained on synthetic fingerprint images.

### Table I

| Name        | # Train Images | # Test Images | PA types                                      | Sensors       |
|-------------|---------------|---------------|-----------------------------------------------|---------------|
| LivDet 2009 [33] | 1000 (1000)   | 1000 (1000)   | Ecoflex, Gelatine, Latex, Modasil, WoodGlue   | Biometrica ItaIdata |
| LivDet 2011 [34] | 1000 (1000)   | 1000 (1000)   | Gelatine, latex, PlayDoh, Silicone, Wood Glue, Ecoflex | DigitalPersona ItaIdata Sagem |
| LivDet 2013 [31] | 1000 (1000)   | 1000 (1000)   | Body Double, Latex, PlayDoh, Wood Glue, Gelatine, Ecoflex, Modasil | Biometrica ItaData CrossMatch Swipe |
| LivDet 2015 [35] | 1000 (1000)   | 1000 (1000)   | Ecoflex, Gelatine, Latex, Liquid Ecoflex, RTV, WoodGlue, Body Double, PlayDoh, OOMOO | Biometrica DigitalPersona GreenBit CrossMatch |
| LivDet 2017 [36] | 1000 (1200)   | 1700 (2040)   | Body Double, Ecoflex, Wood Glue, Gelatine, Latex, Liquid Ecoflex | GreenBit Orcanthus DigitalPersona |
| LivDet 2019 [37] | 1000 (1000)   | 1020 (1224)   | Body Double, Ecoflex, Wood Glue, Gelatine, Latex, Liquid Ecoflex | GreenBit Orcanthus DigitalPersona |
| LivDet 2021 [38] | 1510 (1473)   | 2500 (2460)   | Body Double, Ecoflex, Wood Glue, Gelatine, Latex, Liquid Ecoflex | GreenBit Orcanthus DigitalPersona |
| MSU PPAD [39] | 2500 (3000)   | 2250 (2250)   | Ecoflex, PlayDoh, 2D Matte Paper, 2D Transpareny | CrossMatch |
| MSU PPAD v2 [40] | 4743 (4912)   | 1000 (leave-one-out) | 2D Printed Paper, 3D Universal Targets, Conductive Ink on Paper, Dragon Skin, Gelatine, Gold Fingers, Latex Body Paint, Monster Liquid Latex, PlayDoh, Silicone, Transparency, Wood Glue | CrossMatch |

1 The dataset release agreement for all LivDet databases can be found at [https://livdet.org/registration.php](https://livdet.org/registration.php).
2 Similarly, the dataset release form for the MSU PPAD dataset can be found at [http://biometrics.cse.msu.edu/Publications/Databases/MSU_PPAD/](http://biometrics.cse.msu.edu/Publications/Databases/MSU_PPAD/).
• **Real PA** fingerprint image: fingerprint images captured from a presentation attack artefact on a fingerprint reader.
• **Synthetic bona fide** fingerprint image: synthetic images that mimic the distribution of fingerprint images captured from a real human finger.
• **Synthetic PA** fingerprint image: synthetic images that mimic the distribution of fingerprint images captured from presentation attack artefacts.

We validate the realism of our synthetic bona fide and PA images through extensive qualitative and quantitative metrics including NFIQ2 [32], minutiae statistics, match scores from a SOTA fingerprint matcher, and T-SNE feature space analysis showing the similarity of real bona fide and PA embeddings to the embeddings of our synthetic bona fide and PA fingerprints. Besides verifying the realism of our synthetic PA generator, we also show how SpoofGAN fingerprints can be used to train a DNN for fingerprint PAD. We show this by improving the performance of a PAD model by augmenting an existing fingerprint PA dataset with additional samples from our synthetic generator. We also open the door to jointly optimizing for fingerprint PAD and recognition in an end-to-end learning framework with our ability to generate a large-scale dataset of multiple impressions per finger of both bona fide and PA examples.

More concisely, the contributions of this research are as follows:
• A highly realistic plain print synthetic fingerprint generator capable of generating multiple impressions per finger.
• The first, to the best of our knowledge, synthetic fingerprint PA generator which is capable of producing synthetic representations of both bona fide and PA impressions of the same finger. This opens the door to joint optimization of fingerprint PAD and recognition algorithms.
• Quantitative and qualitative analysis to verify the quality of our generated bona fide and PA fingerprints.
• Experiments showcasing improved fingerprint PAD on both seen and unseen PA material types when augmenting existing fingerprint datasets with our synthetic bona fide and PA fingerprints.
• We release our code and a database of SpoofGAN images to encourage further research in this area https://github.com/groszste/SpoofGAN.

II. RELATED WORK

A. Fingerprint Presentation Attack Detection

One significant risk to the security of fingerprint recognition systems is that of presentation attacks, defined by the international standard ISO/IEC 30107-1:2016 as a “presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system” [41]. The most common type of presentation attacks are spoof attacks, i.e., physical representations of finger-like structures aimed at either mimicking the fingerprint ridge-valley structure of another individual or subverting the user’s own identity.³

Spoof attacks may come in many different forms and materials such as those shown in Figure 1.

Several hardware and software-based solutions to detecting spoof attacks have been proposed. Hardware-based solutions include specialized sensors that leverage various “liveness” cues at the time of acquisition, such as conductivity of the material/finger, sub-dermal imaging, and multi-spectral lighting [42], [43], [44], [45], [46], [47]. On the other hand, software-based solutions typically rely on only the information captured in the grayscale image acquired by the fingerprint reader [39], [48], [49], [50], [51], [52], [53]. Despite the limited publicly available fingerprint PA data, many of the state-of-the-art software-based solutions to fingerprint PAD leverage convolutional neural networks to learn the decision boundary between bona fide and PA images. Some researchers have proposed training their algorithms on smaller patches of the fingerprint images as a way to deal with limited amounts of available training data, which roughly increases the number of training images by a factor proportional to the number of patches [39]. However, given the increased scrutiny over privacy concerns related to biometric datasets, it is not certain whether any PA fingerprint datasets will remain available in the future. Thus, motivating the need for synthetic data to fill this gap.

Another challenge related to limited training data is that of unseen PAs, or fingerprint images arising from never before seen PA instruments. This problem is also commonly referred to in the literature as cross-material generalization. Some strategies proposed to improve the cross-material performance of PA detectors include learning a tighter boundary around the bona fide class via one-class classifiers [54], [55], incorporating adversarial representational learning to encourage robustness to varying material types [56], [57], or applying style transfer to mix textures from some known PA materials to better fill the space of unknown texture characteristics that may be encountered [58], [59]. Similar ideas may apply to synthetic data generation, where new material types can be synthesized by mixing characteristics of known PAs.

B. Synthetic Fingerprint Generation

Research on synthetic fingerprint generation began in the early 2000s with the introduction of SFinGe [60]. Since then, many subsequent works have followed utilizing either hand-crafted approaches [26], [27], [61], learning-based approaches [16], [17], [19], [20], [21], [24], [28], [29], or a combination of both [15], [18], [19]. Classical methods, such as SFinGe, are useful for many applications due to the controllable nature of the generation process; however, still lack the level of realism needed to close the domain gap to real fingerprint images. On the other hand, more recent learning-based approaches have generated increasingly realistic fingerprint ridge patterns but could not generate multiple impressions of the same finger. Fortunately, hybrid approaches, such as [15] and [18], incorporate domain knowledge into the learning-based generation process and can generate high quality fingerprints with multiple impressions per finger. Motivated by the success of hybrid approaches, we also employ a similar architecture as [15] to generate multiple, realistic
fingerprints of each finger. However, unique to our approach, we also have the ability to simulate realistic PA impressions for each finger in a variety of different PA artefact “styles”. To the best of our knowledge, this is the first study on synthetic PA fingerprint image generation.

III. PROPOSED SYNTHETIC PRESENTATION ATTACK FINGERPRINT GENERATOR

In this section, we detail the process of generating synthetic bona fide and PA fingerprints. Motivated by the success of previous multi-stage fingerprint generation methods (e.g., [15], [18], [25]), SpoofGAN generates highly realistic fingerprints in multiple stages. First, unique fingers are synthesized through generating binary master fingerprints which define the fingerprint ridge structure of the finger. Following the master print synthesis stage, perturbations such as random rotation, translation, and non-linear deformation are applied to simulate realistic, repeat impressions. Finally, each generated fingerprint impression is input to a second neural network to impart realistic textures which mimic a database of real fingerprints. An overview of the entire process is given in Figure 2.

A. Master Print Synthesis

The first step in generating synthetic fingerprints with SpoofGAN is generating binary master fingerprints \( I_{id} \in \mathbb{R}^{256 \times 256} \) from a random vector \( z_{id} \in \mathbb{R}^{256} \) sampled from a standard normal distribution (i.e., \( z_{id} \sim \mathcal{N}(0, 1) \)). In particular, we used a standard BigGAN [62] architecture for this task, consisting of a generator \( G_{id} \) and a discriminator \( D_{id} \). Since many PA impressions can exhibit non-realistic fingerprint ridge structures, either from artefacts introduced in the fabrication (e.g., bubbles in the ridges) or in the presentation process (e.g., smudges due to the high elasticity of some PA material types), we chose to train \( G_{id} \) using a database of only bona fide fingerprint impressions consisting of 38,164 images captured on a CrossMatch Guardian200 fingerprint reader. As we will show later, these artefacts for the PA impressions can be introduced in the later texture rendering stage of our synthesis pipeline. The network is trained via an adversarial loss shown in Equation 1, where \( I \) is a binary fingerprint image extracted from a real fingerprint image using the Verifinger v12.0 SDK.

\[
\mathcal{L}_{adv}(G, D) = \mathbb{E}_I [\log D(I)] + \mathbb{E}_z [\log(1 - D(G(z)))] \tag{1}
\]

B. Generating Multiple Realistic Impressions

To generate multiple impressions from a single master print, we apply realistic rotation, translation, and non-linear deformation for each subsequent impression. Rotations are applied via a uniform random sampling in the range \([-30^\circ, 30^\circ]\), whereas translations in both the x and y directions are uniformly sampled in the range \([-25, 25]\) pixels. Finally,
realistic non-linear deformations are applied via a learned, statistical deformation model proposed by Si et al. in [63]. The distortion parameters were learned from a database of 320 distorted fingerprint videos in which the minutiae locations in the first and last frames were manually labeled and the displacements between corresponding minutiae points were used to estimate a distortion field via a Thin-Plate-Spline deformation model [64]. The distortion fields were condensed into a subset of eigenvectors, $e_i$, computed from a Principal Component Analysis of the covariance matrix estimated from the 320 videos. By varying the coefficients, $c_i$, multiplied to each of the eigenvectors, we vary the magnitude of distortion, $d$, applied to an input fingerprint along realistic distortion directions according to equation 2, where $\bar{d}$ are the eigenvalues of each eigenvector and $\bar{d}$ is the average distortion field. For our implementation, we randomly sample the coefficients of the two largest eigenvectors from a normal distribution with mean 0 and standard deviation of 0.66, which were empirically determined to produce reasonable distortions.

$$d \approx \bar{d} \sum_{i=1}^{t} c_i \sqrt{\bar{d}_i e_i}$$ (2)

C. Texture Rendering

The final stage of our fingerprint generation process consists of imparting each fingerprint with a realistic texture that mimics the distribution of real bona fide and PA images. For the generator, $G_t$, we use an encoder-decoder architecture which translates an input, warped binary image, $I_w$, into a realistic fingerprint impression, $I_t$. To promote diversity in the rendered images, a random texture vector sampled from a standard normal distribution (i.e., $z_t \sim \mathcal{N}(0, 1)$) is injected into the network and encoded into $\gamma$ and $\beta$ parameters for performing instance normalization on the intermediate feature maps of $G_t$. Finally, the discriminator, $D_t$, utilizes the same architecture used in the binary master print synthesis network.

The goal of our texture renderer is two fold: i.) generate realistic texture details and ii.) maintain the fingerprint ridge structure (i.e., identity) of the rendered fingerprint between corresponding impressions of the same finger. Thus, we introduce two losses in addition to the conventional GAN loss (eq. 3) to maintain the ridge structure of textured fingerprints. The first is an identity loss to minimize the $L_2$ distance between feature embeddings of corresponding fingerprint impressions using a SOTA fingerprint matcher DeepPrint [9] (eq. 4) and the other is an $L_2$ pixel loss between ground truth binary images and binary images extracted from the textured fingerprints (eq. 5). The $L_2$ pixel loss is computed on the binary images, rather than the grayscale images, to allow for the network to generate diverse “styles” in the generated fingerprints to simulate different pressure, moisture content, and contrast in subsequent impressions; all of which would lead to slightly different loss values compared to the ground truth image unless first converted to binary ridge images. To make the process of binarization of the generated fingerprints differentiable, we train a convolutional autoencoder to binarize input fingerprints which is trained on 38,164 grayscale/binary image pairs. The overall losses for $G_t$ and $D_t$ are given in equations 6 and 7, respectively.

1) GAN loss: Classical min-max GAN loss between the discriminator, $D_t(\cdot)$, trying to classify each original fingerprint image, $I$, as real and each synthetic fingerprint $I_t = G_t(I_w)$ as fake. Meanwhile, $G_t(\cdot)$ is trying to fool $D_t(\cdot)$ into thinking its outputs come from the original image distribution.

$$\mathcal{L}_{adv} = \mathbb{E}_I [\log D(I)] + \mathbb{E}_{I_w} [\log(1 - D(G(I_w)))]$$ (3)

2) DeepPrint loss: $L_2$ distance between the DeepPrint embedding, $R$, extracted from the ground truth grayscale image and the DeepPrint embedding, $\hat{R}$, extracted from the synthesized grayscale fingerprint image.

$$L_{dp} = \frac{1}{2} \sum (R - \hat{R})^2$$ (4)

3) Image/pixel loss: $L_2$ loss between the ground truth binary fingerprint image, $I_w$, and synthesized binary fingerprint image, $\hat{I}_w$.

$$L_i = \frac{1}{2} \sum_{x,y} (I_w(x, y) - \hat{I}_w(x, y))^2$$ (5)

4) Overall loss for $G_t(\cdot)$: $\lambda_1 = 1, \lambda_2 = 2$, and $\lambda_3 = 10$ (determined empirically).

$$\mathcal{L}_{G_t} = \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{dp} + \lambda_3 \mathcal{I}_i$$ (6)

5) Overall loss for $D_t(\cdot)$:

$$\mathcal{L}_{D_t} = \mathcal{L}_{adv}$$ (7)

Unlike the binary master print synthesis and warping stages, an individual texture rendering network is trained for each material type (bona fide, ecoflex PA, PlayDoh PA, etc.). Due to the limited number of images in our PA dataset, we pretrained a texture rendering network on the 282K unique fingerprint database taken from the MSP longitudinal database introduced in [3].^4 Initially, following the pretraining, two texture rendering networks are trained further, one on the dataset of 38,164 bona fide only impressions and the other on the 3,366 PA fingerprint images consisting of all PA types aggregated together. Finally, we further finetune the model trained on all PAs for each of the individual PA types to give more fine-grained control on the specific PA style being generated. Thus, unlike the binary master print synthesis and warping stages which are shared, each individual PA type has its own rendering network. Alternatively, a conditional GAN structure could be used to generate PA classes of each type within a single network; however, we found that due to the very limited amount of training images in some PA types (e.g., 50 images), finetuning for just a few epochs on each PA type individually produced higher quality images.

[^4]: This database is not publicly available, but the pretrained model can be made available upon request.
TABLE II
ARCHITECTURE FOR $G_{id}()$ ($Ch = 48$)

| Layer                          | Output Dim. |
|-------------------------------|-------------|
| 0. Input                      | 512         |
| 1. ReLU(Dense)                | 12, 288     |
| 2. Reshape                    | 4 x 4 x 16c |
| 3. ResBlock Up$^2$            | 8 x 8 x 16c |
| 4. ResBlock Up$^2$            | 16 x 16 x 8c|
| 5. ResBlock Up$^2$            | 32 x 32 x 8c|
| 6. ResBlock Up$^2$            | 64 x 64 x 4c|
| 7. ResBlock Up$^2$            | 128 x 128 x 2c|
| 8. Self Attention             | 128 x 128 x 2c|
| 9. ResBlock Up$^2$            | 256 x 256 x 4c|
| 10. Tanh(Conv2d(ReLU(Batch Norm))) | 256 x 256 x 1 |

† Layer contains conditional batch norm.

TABLE III
ARCHITECTURE FOR $D_{id}()$ ($Ch = 48$)

| Layer                          | Output Dim. |
|-------------------------------|-------------|
| 0. Input                      | 256 x 256 x 1 |
| 1. ResBlock Down              | 128 x 128 x 1 |
| 2. ResBlock Down              | 64 x 64 x 2c |
| 3. Self Attention             | 64 x 64 x 2c |
| 4. ResBlock Down              | 32 x 32 x 4c |
| 5. ResBlock Down              | 16 x 16 x 8c |
| 6. ResBlock Down              | 8 x 8 x 8c   |
| 7. ResBlock Down              | 4 x 4 x 16c  |
| 8. ResBlock                  | 4 x 4 x 16c   |
| 9. Dense(Global Sum Pooling(ReLU)) | 1          |

D. Training Details

For training the master print generator, $G_{id}$, an Adam optimizer with a learning rate of 0.0001 was used, whereas a Moving Average Optimizer with an initial learning rate of 0.0004 was used to train the discriminator, $D_{id}$. Furthermore, the master print generator was trained with a batch size of 8 on two NVIDIA GeForce RTX 2080 Ti GPUs for a total of 178 epochs, where each epoch contained 100 batches. To help balance the training, the generator was updated twice for every update of the discriminator. The architecture for both $G_{id}$ and $D_{id}$ are given in Table II and Table III, respectively.

Finally, $G_{t}$ and $D_{t}$ of the texture renderer utilized the same optimizers as $G_{id}$ and $D_{id}$, respectively; however, $G_{t}$ was updated 3 times for every update of $D_{t}$. To increase the diversity in the generated samples, multiple checkpoints of the texture renderer are used in generating the synthetic data that was used for training the PAD model. The full architectures for $G_{id}$ and $D_{id}$ are given in Tables IV and V, respectively. For completeness, the architecture for the CNN-based fingerprint binarizer used in training the texture render is given in Table VI and the architecture for the texture encoder (which encodes $\gamma$ and $\beta$ parameters from a random texture vector) utilized in $G_{t}$ is given in Table VII.

IV. EXPERIMENTAL RESULTS

In this section, we aim to validate the realism of our synthetic bona fide and PA images via several qualitative and quantitative experiments. First, we provide details on the datasets involved in the following experiments, followed by some example fingerprint images generated by SpoofGAN to qualitatively compare with real fingerprint images. Finally, several quantitative metrics are used to compare the utility and distribution of SpoofGAN generated fingerprint images compared to the real fingerprint images.

A. Datasets

A main motivation for this paper is the lack of large-scale, publicly available fingerprint PAD datasets. Some of
the largest datasets that are available have resulted from the biennial LivDet competition series dating as far back as 2009 [31], [34], [35], [36], [37], [38]. A more comprehensive list of the fingerprint PA datasets currently available to the research community is given in Table I, whereas the datasets used in this paper are given in Table VIII. In this paper we focus our experiments on fingerprint images obtained via the CrossMatch optical reader from LivDet 2013, LivDet 2015, and the Government Controlled Test (GCT) dataset of bona fide and PA fingerprints collected as part of the IARPA ODIN program. Our training dataset for SpoofGAN, referred to as GCT 1-5, consists of 38,164 bona fide fingerprint images and 3,366 PA fingerprint images from 2,007 fingers and 11 different PA types (Dragon Skin, Ecoflex, Paper, Silicone, Transparency, Gelatine, Glue, PDMS, Knox Gelatine, Gummy Overlay, and Tattoo). For our evaluations, we have followed the same train/test protocol referenced in LivDet 2015 and LivDet 2013, as well as reserved a fraction of the GCT dataset, GCT 6, as an evaluation dataset.

### Table VII

| Layer         | Output Dim. |
|---------------|-------------|
| 0. Input      | 128         |
| 1. ReLU(Dense) | 128         |
| 2. ReLU(Dense) | 128         |
| 3. ReLU(Dense) | 128         |
| 4. ReLU(Dense) | 128         |

generated fingerprints compared to a real fingerprint dataset, NFIQ2 quality scores, match score distributions from a SOTA fingerprint matcher, and identity leakage experiments.

1) Presentation Attack Detection Performance of Real Vs. Synthetic Fingerprints: Our first evaluation to verify the quality of our synthetic bona fide and PA fingerprints is to see whether a pretrained PAD algorithm trained on similar, real fingerprints performs equally well on our synthetic fingerprints. In particular, we pretrained an Inception v3 network on the GCT 1-5 data to classify between bona fide and PA fingerprint samples. Then, we evaluated the PAD performance on LivDet 2015 CrossMatch images compared to an equivalent sized database of synthetic fingerprints. As shown in Table IX, the attack presentation classification error rate (APCER), computed at threshold corresponding to bona fide classification error rate (BPCER) of 0.2%, is similar across multiple PA types of both datasets, supporting our hypothesis that the synthetic samples should be useful in training additional PAD models without access to a large database of real bona fide and PA fingerprints for training. Lastly, the embeddings of both real and synthetic images in the T-SNE embedding space suggest high similarity between the embeddings of real and synthetic images (see Figure 5).

2) Feature Similarity Between Real and Synthetic Fingerprints: For synthetic fingerprint images to be useful as a substitute for real fingerprint images, the features between a database of real fingerprint images and synthetic fingerprint images should closely align. For this analysis, we computed several statistics from the LivDet 2015 CrossMatch training dataset (bona fides only) and 1,500 SpoofGAN bona fide fingerprint images which are shown in Table X. In terms of fingerprint area, SpoofGAN images are, on average, smaller compared to the real fingerprint database. Since our training dataset consists of images of all 10 fingers, there is a bias toward smaller fingerprints considering the thumb as a minority class. Given this assumption, it is perhaps unsurprising that a GAN-based generation approach might exaggerate this class imbalance and generate smaller fingerprint area impressions. This problem is related to mode-collapse and has been noted in several GAN related works [65], [66], [67], with some recent papers proposing strategies to improve the generation process in class-imbalanced datasets [68], [69].

Next, we computed the NFIQ 2.0 quality metric [32] on both datasets (see Figure 6). The NFIQ 2.0 scores for SpoofGAN are, on average, lower compared to LivDet. However, since one of the features considered in NFIQ 2.0 is the fingerprint area, we recomputed the scores on a 256 × 256 center crop of each of the fingerprints and observed that independent of fingerprint area, the NFIQ scores between SpoofGAN images and LivDet are much more aligned (36.88 ± 10.18 vs. 43.65 ± 14.12).

Lastly, we computed some additional metrics specific to the distribution of minutiae since many of the state-of-the-art fingerprint algorithms incorporate minutiae information.

5We selected CrossMatch for our experiments since it is one of the most popular slap (4-4-2) capture readers used in law enforcement, homeland security and civil registry applications.

6APCER and BPCER are the standard metrics according to ISO/IEC 30107-1:2016; however, other metrics have also been reported in the literature, including true detection rate at a fixed false detection rate, where BPCER=FDR and APCER=1-TDR (albeit, APCER is computed per PAI).
TABLE VIII
SUMMARY OF THE PA DATASETS USED IN OUR EXPERIMENTS

| Dataset                  | LivDet 2013 | LivDet 2015 | GCT 1-5          | GCT 6          | MSU FPAdv2
|--------------------------|-------------|-------------|------------------|---------------|------------
| Fingerprint Reader       | CrossMatch  | CrossMatch  | CrossMatch       | CrossMatch    | CrossMatch |
| Model                    | L. Scan Guardian | L. Scan Guardian | Guardian200 | Guardian200 | Guardian200 |
| Resolution (dpi)         | 500         | 500         | 500              | 500           | 500        |
| # Bona Fide Images (Train/Test) | 1,250 / 1,250 | 1,510 / 1,500 | 38,164 / 0     | 7,357 / 14,236 | 4,743 / 1,000 |
| # PA Images (Train/Test) | 500 / 440   | 1,473 / 1,448 | 3,366 / 0       | 2,550 / 1,829 | 4,912      |
| PA Materials             | Body Double, Latex, PlayDoh, Wood Glue | Ecoflex, Gelatine, PlayDoh, OOMOO Body Double | Dragon Skin, Ecoflex, Knox Gelatine, Silicone, Transparency, Gelatine, Glue, PDMS, Tattoo Paper, Gummy Overlay | Ecoflex, Silicone, Gummy Overlay, Tattoo, Knox Gelatine |

† There are 4,912 total PA images but the number of train/test images depends on which PA is left out for the cross-material generalization evaluation.

TABLE IX
PAD MODEL TRAINED ON REAL FINGERPRINTS (GCT 1-5) AND EVALUATED ON LIVDET 2015 CROSSMATCH IMAGES (TOP ROW) AND AN EQUIVALENT SIZED SYNTHETIC FINGERPRINT DATASET (BOTTOM ROW)†. RESULTS GIVEN IN APCER AT A THRESHOLD CORRESPONDING TO BPCER = 0.2%

|                  | BodyDouble | Ecoflex | PlayDoh | OOMOO | Gelatine |
|------------------|------------|---------|---------|-------|----------|
| Real             | 0%         | 0.43%   | 1.42%   | 0.42% | 0%       |
| Synthetic        | 0%         | 0%      | 0%      | 0.42% | 0.88%    |

† There are 1,500 bona fide and 1,448 PA fingerprint test images for CrossMatch in LivDet 2015, which we have replicated with synthetic data. Specifically, the PA images consist of 300 Body Double, 270 Ecoflex, 297 OOMOO, 281 PlayDoh, and 300 Gelatine images.

The average minutiae count and minutiae quality computed by Verifinger 12.0 are given in Table X. The average number of minutiae found in SpoofGAN seems to be lower compared to the CrossMatch images from LivDet 2015; however, the minutiae per Megapixel is similar for both datasets (59.74 vs. 59.49). The minutiae quality given by Verifinger is also very similar between the two datasets (71.89 vs. 70.78).

3) Diversity in the Generated Fingers: To verify that SpoofGAN generated fingerprints mimic the similarity score distribution of real bona fide and PA fingerprints, we have computed genuine and imposter matches with Verifinger v12.0. In particular, we computed match scores (genuine and imposter) between the bona fide impressions of the LivDet 2015 CrossMatch images as well as between the bona fide samples of synthetic SpoofGAN fingerprints. These distributions are shown in Figure 7 (a). This figure highlights that SpoofGAN is generating diverse fingers with similar intraclass and interclass variation as the real LivDet 2015 CrossMatch dataset, albeit producing slightly lower genuine scores compared to the real dataset. However, in terms of recognition performance at a fixed false acceptance rate (FAR), the performance between the two datasets is quite similar, despite the slightly shifted genuine distribution of the SpoofGAN images (see Table XI).

Furthermore, we computed genuine score distributions between the individual PA types generated by SpoofGAN and their corresponding bona fide impressions. These distributions are given in Figure 7 (b). Here we see that Verifinger is able to successfully match PA and bona fide fingerprint images belonging to the same finger, which we believe opens the door to synthesising a large-scale PA and recognition dataset that can be used to train and evaluate joint PAD and recognition algorithms.

4) Identity Leakage: A major advantage of generating synthetic fingerprint data is that, theoretically, no fingerprint ridge structure matches that of an actual user in the training database. However, there remains a concern that synthesis methods, such as GANs, may inadvertently over-fit and leak private information from the training corpus [70]. Therefore, it is instructive to investigate whether, and to what degree, any of our SpoofGAN generated images are revealing, i.e., match with sufficient confidence, the identities present in our training database. Toward this end, we have computed match scores between 1,500 SpoofGAN generated bona fide fingers and each of the 38,164 real bona fide fingers in our training set. Out of the roughly 57.2 million (1,500 x 38,164) potential matches, only 50 comparisons exceeded the matching threshold of 48 set by Verifinger for a false acceptance rate of 0.01% with a maximum match score of 81. Furthermore, the 50 matches resulted from just 29 SpoofGAN generated fingers.
Fig. 3. Example real bona fide and PA images and synthetic bona fide and PA images generated by SpoofGAN of various material types: (a) bona fide, (b) ecoflex, (c) body double, (d) conductive ink on paper, (e) tattoo, and (f) gelatine.

out of the 1,500 evaluated. Some example matched SpoofGAN and real fingerprint image pairs are shown in Figure 8 along with their corresponding match scores.

As part of future work, the risk of identity leakage could be further mitigated with either i.) a larger training database to avoid the network from simply memorizing training samples, ii.) performing an identity check with the training database upon generation of each fingerprint to trigger a re-sampling if matched with an existing finger (though potentially an expensive operation), or iii.) composing the master print pattern
Fig. 4. Example images of multiple impressions of the same finger generated by SpoofGAN. (a) and (b) show three impressions each of two fingers rendered in a bona fide style; whereas (c) and (d) show three impressions each of the same two fingers in a PA style (ecoflex and body double, respectively).

TABLE X

| Measure                     | LivDet 2015 CrossMatch | SpooGAN |
|-----------------------------|-------------------------|---------|
|                             | Mean  | Std. Dev. | Mean  | Std. Dev. |
| Total Minutiae Count        | 55.56 | 18.43     | 40.45 | 11.57     |
| Ridge Ending Minutiae Count | 30.58 | 12.12     | 20.99 | 6.67      |
| Ridge Bifurcation Minutiae Count | 24.98 | 9.18    | 19.46 | 6.61      |
| Verifinger Minutiae Quality | 71.89 | 16.00     | 70.78 | 15.15     |
| Fingerprint Area (Megapixels) | 0.93  | 0.30      | 0.68  | 0.15      |
| Fingerprint Image Quality (NFIQ2) | 60.18 | 19.35     | 44.34 | 14.38     |

TABLE XI

| FAR | LivDet | SpooGAN |
|-----|--------|---------|
| 0.01% @ threshold=48 | 99.87% | 100%    |
| 0.001% @ threshold=60 | 99.80% | 100%    |
| 0.0001% @ threshold=72 | 99.74% | 99.80%  |
| 1e-05% @ threshold=84 | 99.41% | 99.47%  |
| 1e-06% @ threshold=96 | 99.35% | 98.87%  |

D. Improved Presentation Attack Detection With Synthetic Fingerprints

Ultimately, our synthetic fingerprints should offer some utility in advancing the training of fingerprint PAD algorithms. Toward this end, we have augmented three existing, publicly available datasets of bona fide and PA fingerprints with our synthetic fingerprints in an effort to improve the performance beyond that achievable when training on the real bona fide and PA images from each dataset alone. For this evaluation, we have trained several PAD models on the following training set compositions: i.) synthetic bona fide and PA images only, ii.) real bona fide and PA images only, iii.) synthetic bona fide and PA images plus only real bona fide images, and iv.) synthetic bona fide and PA images plus real bona fide and PA images. We used the SpoofBuster model, which consists of two Inception v3 networks, one trained on the whole image input and the other trained on 96 × 96 minutiae centered patches [39]. The final PA score is the weighted fusion of the two networks, with a minutiae patch score weight of 0.8 and a whole image score weight of 0.2. Each of the models are trained from scratch using Tensorflow on a single Nvidia TitanX 1080 GPU on their respective datasets with identical hyper-parameters (learning rate of 0.01, step decay learning rate schedule, adam optimizer with default parameters, and total training updates of 200,000 steps).
Shown in Table XII, we have evaluated each of the models on their respective test sets. Despite the lower performance when training on synthetic data alone compared to training on real data, we see improvement in the overall PA classification performance when the real training data is augmented with samples from our synthetic bona fide and PA generator. For example, the error of the minutiae patch model trained on real data from LivDet 2013 is reduced by 91.03% (from 15.60% to 1.40%) when augmented with synthetic data. Similarly, the error on LivDet 2015 is reduced from 0.48% to 0.0%, where the error on GCT 6 remained the same at 0.0%.

Interestingly, as seen in Table XIII, we also see substantial improvements in cross-material generalization when incorporating synthetic SpoofGAN images into the training dataset of our PAD model. Specifically, we compared the cross-material generalization of a PAD model trained on LivDet 2013 with the same PAD model trained on LivDet 2013 augmented with SpoofGAN images. We observe drastic reduction in APCER across 7 different PA types from the MSU FPADv2 dataset which are not included in LivDet 2013 training dataset.
nor were these PA types replicated in the synthetic images obtained from SpoofGAN that were used to augment the training dataset. Overall, the average APCER reduced from 76.71% to 54.03% due to the addition of SpoofGAN training images.

Furthermore, Figure 9 shows the trend in performance on LivDet 2015 as we keep the number of synthetic training samples fixed but varying the percentage of real data included when training the whole image-based PAD model. This figure suggests that when augmenting the training set with synthetic data, just 25% of the original (real) data is required to obtain similar performance to training on 100% of the real data alone, which significantly reduces the time and resources required for data collection to obtain similar performance. In fact, the behavior of the real training data curve (shown in blue) in the early stages (e.g., 5% and 10% of the total training data) exhibits a very sharp decrease in APCER, suggesting that many of the PA vs. bona fide features can be learned from very few samples; however, the subtle features that give the PAD model the last 10% of improvement require 3 or 4 times as many data samples. Importantly, from the real plus synthetic curve (shown in orange), the additional data required to gain that extra 10% can instead be generated by SpoofGAN.

Lastly, even if researchers and practitioners have access to a large, private database of real bona fide fingerprint images, collecting an equivalent database of PA fingerprints is more difficult and costly; therefore, they may wish to augment their database of real bona fide images with synthetic PA images. As seen in Table XII, mixing synthetic data with only real bona fide fingerprint examples improves the performance over training on just the synthetic examples alone; however, synthetic PA data is still not a substitute for collecting real PA examples as the performance still lags quite significantly.
fixed and vary the percentage of real data included in training. Performance
Fig. 9. PAD performance as we keep the number of synthetic training samples
LivDet 2015 dataset.

an online fashion with a PAD network may provide additional
future; thus, instilling the generation process with the ability
diversity of the generated images could be improved. A related
of generating multiple genuine and imposter fingerprints of
match score distributions computed by the state-of-the-art
Additionally, our synthetic fingerprints closely resemble a
achieved when training on real PA fingerprint datasets only,
improving the performance of a PA detector beyond that
sions of both bona fide and PA varieties. We demonstrated
utility of our synthetically generated PA fingerprints in
improving the performance of a PA detector beyond that

V. CONCLUSION AND FUTURE WORK
In this work, we presented a GAN-based synthesis method
for generating high quality 512 × 512 plain fingerprint impres-
sions of both bona fide and PA varieties. We demonstrated the
of various statistics, such as distribution of
minutiae, NFIQ 2.0 image quality, PA classification, and
match score distributions computed by the state-of-the-art
Veriﬁnger 12.0 SDK. Finally, since our method is capable of
generating multiple genuine and impostor fingerprints of
unique ﬁngers in both bona ﬁde and PA types, we open the
doors for large-scale training and evaluation of joint
fingerprint PAD and recognition algorithms; overcoming a current
limitation given the existing scale of publicly available PA
fingerprint datasets.

Despite the demonstrated utility of our synthetic PA images,
there remains several limitations that will be addressed in
future work. First, given the lower standard deviation across
the various ﬁngerprint metrics presented in Table X, the
diversity of the generated images could be improved. A related
issue, speciﬁc to ﬁngerprint PAD, is the ever increasing novelty
of PA materials and types that may be encountered in the
future; thus, instilling the generation process with the ability
to adapt to novel PA types is a promising future research
direction. Lastly, training the synthetic ﬁngerprint generator in
an online fashion with a PAD network may provide additional
supervision to generate more useful bona fide and PA examples
to improve the PAD performance.

REFERENCES
[1] F. Galton, Finger Prints. New York, NY, USA: Macmillan, 1892.
[2] S. Pankanti, S. Prabhakar, and A. K. Jain, “On the individuality of
fingerprints,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 8,
pp. 1010–1025, Aug. 2002.
[3] S. Yoon and A. K. Jain, “Longitudinal study of ﬁngerprint recognition,”
Proc. Nat. Acad. Sci. USA, vol. 112, no. 28, pp. 8555–8560, 2015.
[4] B. Dorizzi et al., “Fingerprint and on-line signature veriﬁcation com-
petitions at ICB 2009,” in Proc. Int. Conf. Biometrics. Berlin, Germany:
Springer, 2009, pp. 725–732.
[5] K. Cao and A. K. Jain, “Fingerprint indexing and matching: An
integrated approach,” in Proc. IEEE Int. Joint Conf. Biometrics (IJCB),
Oct. 2017, pp. 437–445.
[6] D. Song and J. Feng, “Fingerprint indexing based on pyramid deep
convolutional feature,” in Proc. IEEE Int. Joint Conf. Biometrics (IJCB),
Oct. 2017, pp. 200–207.
[7] D. Song, Y. Tang, and J. Feng, “Aggregating minutia-centred deep con-
volutional features for fingerprint indexing,” Pattern Recognit., vol. 88,
pp. 397–408, Apr. 2019.
[8] R. Li, D. Song, Y. Liu, and J. Feng, “Learning global fingerprint features
by training a fully convolutional network with local patches,” in Proc.
Int. Conf. Biometrics (ICB), Jun. 2019, pp. 1–8.
[9] J. J. Engelsma, K. Cao, and A. K. Jain, “Learning a ﬁxed-length
ﬁngerprint representation,” IEEE Trans. Pattern Anal. Mach. Intell.,
vol. 43, no. 6, pp. 1981–1997, Jun. 2021.
[10] C. Lin and A. Kumar, “A CNN-based framework for comparison of
contactless to contact-based ﬁngerprints,” IEEE Trans. Inf. Forensics
Security, vol. 14, no. 3, pp. 662–676, Mar. 2019.
[11] S. A. Grosz, J. J. Engelsma, E. Liu, and A. K. Jain, “C2CL: Contact to
contactless ﬁngerprint matching,” IEEE Trans. Inf. Forensics Security,
vol. 17, pp. 196–210, 2022.
[12] J. J. Engelsma, A. K. Jain, and V. N. Boddeti, “HERS: Homomor-
phically encrypted representation search,” IEEE Trans. Biometrics, Behav.,
Identity Sci., vol. 4, no. 3, pp. 349–360, Jul. 2022.
[13] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang,
“Biometrics recognition using deep learning: A survey,” 2019,
arXiv:1912.00271.
[14] K. Sundararajan and D. L. Woodard, “Deep learning for biometrics: A
survey,” ACM Comput. Surv., vol. 51, no. 3, pp. 1–34, May 2018.
[15] J. J. Engelsma, S. A. Grosz, and A. K. Jain, “PrintGAN: Synthetic
ﬁngerprint generator,” 2022, arXiv:2201.03674.
[16] R. Bouzaglo and Y. Keller, “Synthesis and reconstruction of ﬁngerprints
using generative adversarial networks,” 2022, arXiv:2201.06164.
[17] K. Bahmani, R. Plesh, P. Johnson, S. Schuckers, and T. Swyka, “High
fidelity ﬁngerprint generation: Quality, uniqueness, and privacy,” 2021,
arXiv:2105.10403.
[18] A. B. V. Wyzkowski, M. P. Segundo, and R. de Paula Lemes, “Level
three synthetic ﬁngerprint generation,” in Proc. 25th Int. Conf. Pattern
Recognit. (ICPR), Jan. 2021, pp. 9250–9257.
[19] V. Mistry, J. J. Engelsma, and A. K. Jain, “Fingerprint synthesis: Search
with 100 million prints,” in Proc. IEEE Int. Joint Conf. Biometrics (IJCB),
Sep. 2020, pp. 1–10.
[20] K. Cao and A. Jain, “Fingerprint synthesis: Evaluating ﬁngerprint search
at scale,” in Proc. Int. Conf. Biometrics (ICB), Feb. 2018, pp. 31–38.
[21] M. A.-N. I. Fahim and H. Y. Jung, “A lightweight GAN network for large
scale ﬁngerprint generation,” IEEE Access, vol. 8, pp. 92918–92928,
2020.
[22] M. S. Riazi, S. M. Chavoshian, and F. Koushanfar, “SynFi: Automatic
fingerprint synthesis based on statistical feature models,” in Proc.
IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), Oct. 2018.
[23] P. Bontrager, A. Roy, J. Togelius, N. Memon, and A. Ross, “DeepMaster-
Prints: Generating MasterPrints for dictionary attacks via latent variable
evolution,” in Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst.
(BTAS), Oct. 2018, pp. 1–9.
null