FPGAN-Control: A Controllable Fingerprint Generator for Training with Synthetic Data

Alon Shoshan[1] **Nadav Bhonker**[1] **Emanuel Ben Baruch**[1] **Ori Nizan**[*2]
Igor Kviatkovsky[1] **Joshua Engelsma**[*3] **Manoj Aggarwal**[1] **Gérard Medioni**[1]
[1]Amazon [2]Technion - Israel Institute of Technology [3]Rank One Computing

Abstract

Training fingerprint recognition models using synthetic data has recently gained increased attention in the biometric community as it alleviates the dependency on sensitive personal data. Existing approaches for fingerprint generation are limited in their ability to generate diverse impressions of the same finger, a key property for providing effective data for training recognition models. To address this gap, we present FPGAN-Control, an identity preserving image generation framework which enables control over the fingerprint's image appearance (e.g., fingerprint type, acquisition device, pressure level) of generated fingerprints. We introduce a novel appearance loss that encourages disentanglement between the fingerprint's identity and appearance properties. In our experiments, we used the publicly available NIST SD302 (N2N) dataset for training the FPGAN-Control model. We demonstrate the merits of FPGAN-Control, both quantitatively and qualitatively, in terms of identity preservation level, degree of appearance control, and low synthetic-to-real domain gap. Finally, training recognition models using only synthetic datasets generated by FPGAN-Control lead to recognition accuracies that are on par or even surpass models trained using real data. To the best of our knowledge, this is the first work to demonstrate this.

Figure 1. **Traversing the appearance space of FPGAN-Control.** We present an animation of four fingerprints generated by FPGAN-Control with the following properties: (a) each of four animated fingerprints belongs to a unique synthetic identity which is preserved throughout the animation; (b) at every moment the appearance of each fingerprint is shared; and (c) the shared appearance gradually changes over time. [Animated figure, please view at alonshoshan10.github.io/fpgan_control/].

1. Introduction

Within the past few years the biometric community has shown an increased interest in the use of synthetic data for recognition system development [3,6,13]. This is due to two primary reasons. First, the state-of-the-art tools for photo-realistic image generation have seen a significant leap in terms of image quality [11, 23, 24, 34, 37] and the level of control over the generated output [10, 27, 38, 40]. While the former reduces the synthetic-to-real domain gap, the latter ensures biometric identity uniqueness and preservation under controllable intra-class variations, enabling the use of synthesized data for both training and evaluation. Second, recent privacy and ethical concerns regarding the use of existing datasets [16] have encouraged researchers to consider replacing real biometric data with synthetic data.

While several efforts have seen success in training models using synthetic data in the face recognition domain [3, 6, 32], the usage of synthetic data for training recognition models in the fingerprints domain has started to gain attention only recently. One reason for this might be, that while several fingerprint generators are available [4,7,30,43], they lack the ability to generate different impressions for a newly generated identity. To tackle this issue, PrintsGAN [13], a

[*]Work done while at Amazon.

method for generating novel identities along with their identity preserving variations has been proposed recently. Although very useful, PrintsGAN relies on a complex three-step generation process focusing on a specific type of fingerprint, which limits the appearance variability in a general sense. In addition, the approach lacks the ability to control the generated fingerprint appearance attributed to: fingerprint type (rolled or slap), scanner type, moisture and pressure levels *etc*. Finally, the PrintsGAN generator is not publicly available and only a sample set of 35,000 identities was released.

In this work, we propose an end-to-end GAN based learning scheme for the task of fingerprint image generation, which we name FPGAN-Control (**F**inger**P**rint **GAN**-Control). In particular, our approach enables control over the appearance of the generated fingerprint images while preserving the biometric identity information. We rely on the GAN-control framework [38] developed initially for controllable and identity-preserving face image generation.

While it is intuitive to define controllable facial characteristics (*e.g*., expression, age, hair style) and acquisition properties (*e.g*., orientation, illumination), it is less straightforward for the domain of biometric fingerprints. Thus, to address possible variations in impressions of the same finger, we define a single generic "appearance" property which encompasses many of the possible impression variations. To this end, we propose a novel and interpretable appearance loss to enforce disentanglement between the fingerprint identity and its appearance in the GAN's latent space. In particular, we apply a smoothing kernel and downsample the generated images in each training batch to filter out their high-frequencies while retaining the appearance properties. Then, we encourage similarity between blurred images generated using an identical appearance latent, while separating those generated by different appearance latents. This way, we enable control over the appearance of the generated fingerprints. We visualize the control capabilities of the fingerprint appearance while preserving its identity in Figure 1.

Finally, we use fingerprints generated with FPGAN-Control to train recognition models. We empirically establish that training fingerprint recognition models using only synthetic identities results in accuracy levels that are comparable and even surpassing those of models trained with real data. To the best of our knowledge, our method is the first to achieve this in the fingerprints domain. This allows to avoid relying on sensitive personal data, addressing common privacy and security concerns, highly valued by the biometric community. To facilitate further research, we will release our code and pretrained models.

To summarize, our contributions include:

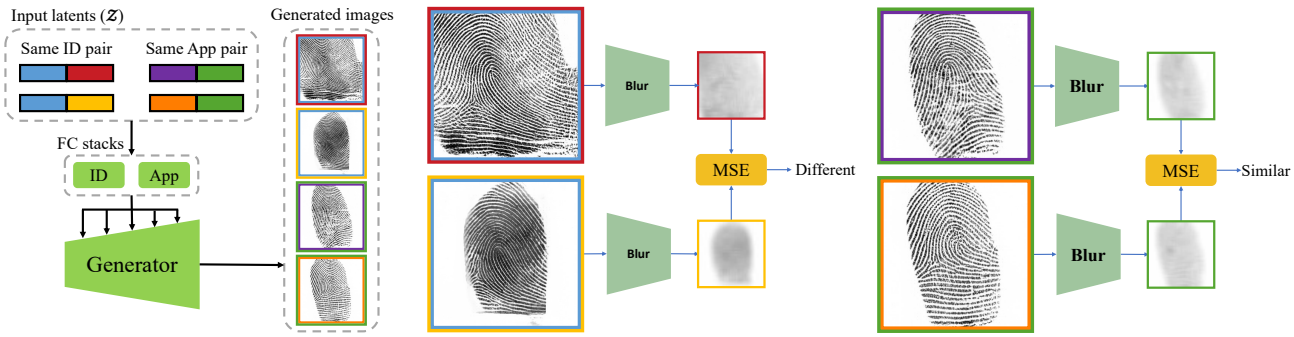
1. We introduce FPGAN-Control, an end-to-end training method for controllable fingerprint image generation. FPGAN-Control is designed with disentangled latent space in mind to allow generation of novel fingerprint identities along with a variety of impressions.
2. We propose an intuitive dedicated appearance loss, operating on the fingerprint image's low frequencies. This loss is crucial for disentangling the generator's latent space.
3. We train recognition models using purely synthetic data, generated by FPGAN-Control, reaching or surpassing the performance of models trained on real data.
4. Our code and models will be publicly released to facilitate privacy preserving research in the domain of fingerprint recognition.

2. Related Work

The scarcity of publicly available fingerprint datasets has led to increased interest in developing methods to synthesize fingerprints. Earlier methods relied heavily on "hand-crafted" solutions based on the available knowledge on fingerprints, *e.g*., they leverage the studied behavior of the friction ridge patterns comprised of interwoven ridges, valleys, minutiae points, and pores [1, 9, 22, 45]. In more recent years, as Generative Adversarial Networks (GANs) [19] have proliferated into a host of photo-realistic image synthesis algorithms, researchers have turned to GANs or "learning based methods" [2, 4, 5, 15, 29, 33, 35], to generate much more realistic fingerprints than the older hand-crafted approaches. Although most of these methods are able to produce high-quality fingerprint images, they suffer from two main deficiencies. First, the methods are only able to generate a single, unique image for each synthetic fingerprint identity. Hence, they are not able to model the intra-class variability. Second, many of these synthesize only patches of fingerprints, rather than full-fingerprints. These limitations motivated methods which introduce control over the generated fingerprints, allowing the generation of multiple impressions per fingerprint identity [13, 20, 31, 42, 43].

These methods usually consist of multiple stages, *e.g*., binary fingerprint generation, distortion, and a GAN to render the binary fingerprint to a realistic fingerprint impression. The need to apply many stages, adds significant complexity to the systems and risk when adapting to new datasets and domains. Most importantly, while all previous work agree that there's a need for synthetic data to conduct research, only two [13, 20] of the above methods evaluate the performance of models trained using their synthetic data. Since Grosz *et al*. [20] proposes a generator of spoof images, the method closest to ours is PrintsGAN [13].

Our method consists of a single training stage based on GAN-Control [38] with a disentangled latent space. We incorporate a novel appearance loss to enable the control of



(a) Training batch generation. (b) Different appearance with same ID. (c) Same appearance with different ID.

Figure 2. **Overview of the proposed FPGAN-Control.** In each training batch (a), both same ID pairs and same appearance pairs are generated. Same ID pairs have the same ID latent vector while same appearance pairs have the same appearance latent vector. The color of the inner image border corresponds to the fingerprint ID and the color of the outer border corresponds to the fingerprint appearance. Each image in the batch is blurred and downsampled, effectively removing it's barometric features while still obtaining many of its appearance features. Blurred images with different appearance latents are pushed one from another (b), while blurred images with the same appearance latent are pulled towards each other (c).

both identity and appearance details. In addition, we do not rely on any hand-crafted image synthesizer or other proprietary algorithms. Our method benefits from the advantages of previous learning-based approaches while not suffering from their deficiencies. In particular, our approach is able to generate multiple impressions of a single finger while preserving its identity; our method consists of a single network trained end-to-end; and training recognition models using synthetic data generated by FPGAN-Control reaches performance comparable to that of models trained on real data.

Worth noting are the recent efforts made towards training face recognition models using synthetic data [3, 6, 25, 26, 28, 32, 44]. For instance, in [32], DiscoFaceGAN [10] was used to generate synthetic face identities for training, and proposed to deploy identity mixup and domain mixup to mitigate the domain gap between real and synthetic images. In [3] the authors released a large-scale synthetic dataset created by rendering 3D face models, containing 1.22M images of 110K identities and utilized it for training. While demonstrating appealing accuracy results for face recognition, it is not trivial to adapt them to the fingerprint domain, as they are specifically handcrafted for the face domain, *e.g.*, using 3D face models [3] or training GANs with domain specific losses [10]).

3. Proposed Approach

In this section we present our framework for training FPGAN-Control, a fingerprint image generation model that allows generating novel biometric identities (IDs) and controlling their appearance variations while preserving the ID. We define the appearance as the image properties that are not related to the fingerprint ID, such as fingerprint type, acquisition device, moisture and pressure levels, *etc*.

3.1. Identity-appearance disentangling

In the design of our approach we rely on the core blocks of GAN-Control [38]. GAN-Control proposes a general two-phase solution for training explicit controllable GANs. In the first phase, we train a disentangled GAN using a set of contrastive losses. As a result, the latent space of the trained GAN is divided into subspaces, each encoding a different image property. The second phase is responsible for enabling explicit control over the generated image by adding property-specific encoders to each subspace of the GAN. While an explicit control over image attributes is useful for content creation, such fine-grained control is not required for the purpose of data generation for recognition model training. Thus, to train FPGAN-Control we adopt only the first phase, resulting in a disentanglement between identity and appearance.

We define the latent space, $\mathcal{Z}$, of FPGAN-Control as a combination of two subspaces $\mathcal{Z}^{id}$ and $\mathcal{Z}^{app}$, associated with the generated fingerprint ID and appearance, respectively ($\mathcal{Z} = \mathcal{Z}^{id} \times \mathcal{Z}^{app}$). Thus, a latent $\mathbf{z} \in \mathcal{Z}$ is the concatenation of the sub-vectors $\mathbf{z}^{id}$ and $\mathbf{z}^{app}$, each of dimension 256. Instead of using a single 8-layered MLP, as done in StyleGAN2 [24], we allocate a separate 8-layered MLP for each subspace. Each training batch contains pairs of same-ID latents (latent vectors with equal identity, $\mathbf{z}^{id}$, and different appearance, $\mathbf{z}^{app}$) and same-appearance latents (equal appearance, $\mathbf{z}^{app}$ and different ID, $\mathbf{z}^{id}$), see Figure 2a. In addition to the StyleGAN2's original adversarial loss, all image pairs are penalized by a weighted combination of contrastive ID and appearance losses (l_{id} and l_{app}):

$$L_c = \sum_{\substack{\mathbf{z}_i, \mathbf{z}_j \in B \\ i \neq j}} l_{id}(\mathbf{z}_i, \mathbf{z}_j) + w_{app} \cdot l_{app}(\mathbf{z}_i, \mathbf{z}_j), \quad (1)$$

where $B = \{\mathbf{z}_i\}_{i=1}^{N_B}$ denotes all latent vectors in the training batch of size N_B. Each of the loss components, l_{id} and l_{app}, has a generic form depending on the corresponding distance function, d_{id} and d_{app}:

$$l_k(\mathbf{z}_i, \mathbf{z}_j) = \begin{cases} \frac{1}{C_k^+} \max\left(d_k(\mathcal{I}_i, \mathcal{I}_j) - \tau_k^+, 0\right), & \mathbf{z}_i^k = \mathbf{z}_j^k \\ \frac{1}{C_k^-} \max\left(\tau_k^- - d_k(\mathcal{I}_i, \mathcal{I}_j), 0\right), & \text{otherwise} \end{cases} \quad (2)$$

where $k \in \{id, app\}$, $\tau_k^{\pm}$ are ID or appearance thresholds associated with same and different sub-vectors and $C_k^{\pm}$ are normalizing constants. The key part of this scheme is in the selection of appropriate distance functions, d_{id} and d_{app}. We define d_{id} as the cosine distance between two embedding vectors extracted by a pre-trained fingerprint recognition model, θ_{id}. Unfortunately, there is no available identity invariant metric for computing the appearance distance, d_{app}, between two images. To address this, we developed an appearance distance focusing on the dissimilarity between a pair of images in the low-frequency domain.

3.2. Appearance distance

Intuitively, given a fingerprint scan, most of the ID-related biometric features are contained in the image's high frequency components while the appearance-related information is contained in its low frequency components. We use this observation in the design of our appearance distance function, d_{app}. Given two fingerprint images $\mathcal{I}_i$ and $\mathcal{I}_j$, we downsample and blur each image using a Gaussian smoothing filter:

$$\tilde{\mathcal{I}} = \text{resize}(\mathcal{I}) * h(\sigma, n), \quad (3)$$

where resize$(\cdot)$ is a bi-linear downsampling operation, h is a Gaussian kernel with variance σ and kernel size n. The purpose of the blur filter is to remove as much of the ID-dependent biometric features as possible from the fingerprint, while preserving the most important appearance-related information. To measure the appearance distance between $\mathcal{I}_i$ and $\mathcal{I}_j$ we compute the pixelwise Mean Squared Error (MSE) between their processed versions:

$$d_{app}(\mathcal{I}_i, \mathcal{I}_j) = \text{MSE}(\tilde{\mathcal{I}}_i, \tilde{\mathcal{I}}_j). \quad (4)$$

In each training batch we use d_{app} to penalize images with different appearance latents but having similar appearance (Figure 2b) and images sharing the same appearance latent but diverging in appearance (Figure 2c). In Section 4 we demonstrate that this intuitive approach is effective in training identity-preserving fingerprint generators with controllable appearance.

3.3. Removing the first sub-sampling layer of fingerprint recognition models

Throughout our experimental work, we observed that training fingerprint recognition models using common off-the-shelf architectures (*i.e.*, ResNet, MobileNet, and EfficientNet) lead to unstable and poor results. We hypothesized that the initial sub-sampling layer of these networks may eliminate valuable fine-grained fingerprint details that are crucial for achieving high discriminative power. Therefore, we removed the initial sub-sampling mechanism from all networks. Specifically, we omitted the first max-pooling layer from ResNet models, and we reduced the stride size of the initial convolutional layer from 2 to 1 in MobileNet and EfficientNet architectures. With this modification, we achieved increased training stability and a significant improvement in test accuracy. We implemented this architectural change in all of the recognition models that were experimented with in the paper.

	R18	R34	R50	R101	M050	M100	Eff-s
w/	82.6	80.3	82.2	83.0	90.8	90.2	90.6
w/o	90.5	89.7	92.0	93.7	93.7	94.2	93.7

Table 1. **Effect of first sub-sampling layer.** TAR@FAR=0.1% results for recognition models trained with and without (w/, w/o) the first backbone's sub-sampling layer. The following backbones where evaluated: ResNet (R), MobileNetV2 (M) and EfficientNetV2 (Eff).

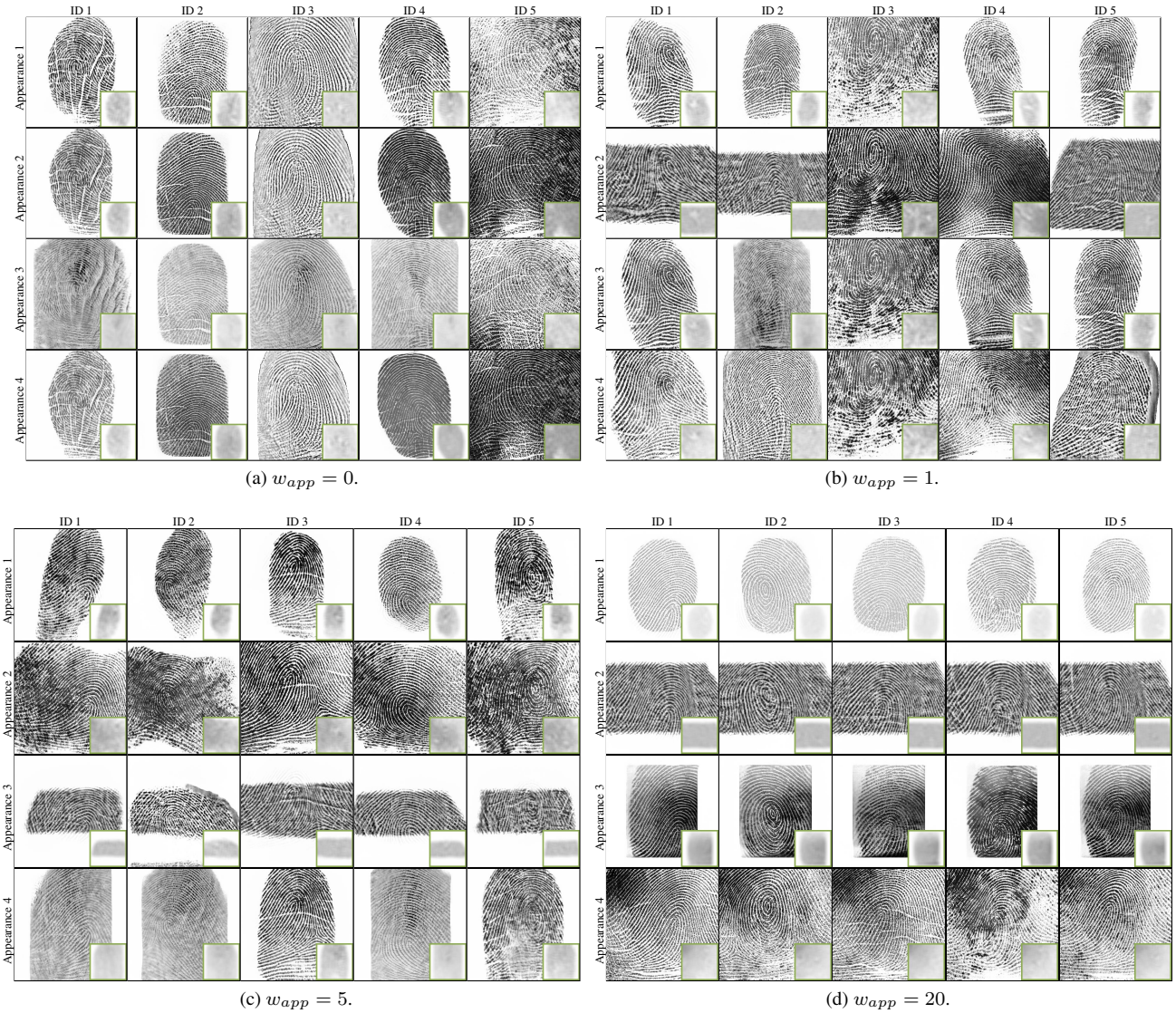
Table 2 presents the results obtained by models trained both with and without the initial sub-sampling layer. As can be seen, training recognition models without the initial sub-sampling layer improves the test accuracy by a large margin for all the tested architectures.

4. Experiments

We first present quantitative and qualitative evaluations of FPGAN-Control, demonstrating its high quality generation, identity preservation and appearance control capabilities. Second, we focus on the ability to train fingerprint recognition models on purely synthetic datasets generated by FPGAN-Control, which is our main goal. We show that, given the multitude of identities and appearance variations that FPGAN-Control can generate, we are able to produce recognition models with higher accuracy even compared to models that were trained with real data.

In our experiments, we used the publicly available NIST N2N dataset (NIST SD 302 [17]). The N2N dataset contains 2,000 unique fingers (of 200 people) with 8 to 15 different impressions per finger. Each finger represents a unique identity. We divided the dataset to 1,600 identities (160 people) for training and 400 (40 people) for testing. The N2N dataset is challenging for the task of training recognition models due to its diverse range of fingerprint images, captured using both traditional and newly developed methods.

We experimented with the following backbones: ResNet18, ResNet34, ResNet50, ResNet101 [21],



(a) $w_{app} = 0$. (b) $w_{app} = 1$.

(c) $w_{app} = 5$. (d) $w_{app} = 20$.

Figure 3. **Generation results of FPGAN-Control trained using different w_{app}.** For a specific FPGAN-Control model, each column represents images generated with the same ID latent vector input and each row represents images generated with the same appearance latent vector input. For visualization of the appearance loss, the small images in green borders show the blurred representation of the fingerprint image used by the loss.

MobileNetV2-050, MobileNetV2-100 [36] and EfficientNetV2-s [39]. As discussed in the previous section, we removed the first sub-sampling operation from all backbones to enhance training stability and improve accuracy. All recognition models were trained using the CosFace loss [41]. During training, we applied random affine transformations.

4.1. Synthetic fingerprint generation

In order to provide a recognition model for FPGAN-Control's identity loss, we first had trained a ResNet18-based model on the 1,600 identities of the training set. We used the same ResNet18 model for training all of our FPGAN-Control models. Note that all our GANs were trained using only the 1,600 identities of the training set, the test set was never used in any form of training. In the following sections, we will provide qualitative and quantitative results that demonstrate the capabilities of FPGAN-Conrol.

4.1.1 Qualitative results

Figure 3 shows qualitative results of four FPGAN-Control versions, each trained with a different appearance loss weight, w_{app} $(0, 1, 5, 20$, where 0 means no appearance loss). Figure 3a demonstrates that without the appearance loss, the appearance variation between images of the same

w_{app}	ID distance	App. distance
0	$0.063_{\pm 0.06}$	$0.026_{\pm 0.04}$
0.25	$0.074_{\pm 0.07}$	$0.027_{\pm 0.04}$
0.5	$0.076_{\pm 0.07}$	$0.033_{\pm 0.04}$
1	$0.122_{\pm 0.08}$	$0.043_{\pm 0.04}$
5	$0.229_{\pm 0.11}$	$0.051_{\pm 0.05}$
20	$0.283_{\pm 0.11}$	$0.057_{\pm 0.05}$
Real data	$0.375_{\pm 0.19}$	$0.058_{\pm 0.05}$

Table 2. **Intra class statistic.** ID distance and App distance correspond to the average recognition distance and the mean appearance distance between two images of the same ID, respectively.

w_{app}	0	0.25	0.5	1	5	20
Dist↓	0.053 ±0.04	0.044 ±0.04	0.038 ±0.03	0.025 ±0.02	0.009 ±0.01	0.002 ±0.00

Table 3. **Appearance control precision vs. appearance loss weight,** w_{app}. For each FPGAN-Control model, we measured the average appearance distance between pairs of images sharing the same appearance latent, but having a diffrent ID latent.

ID is small and mostly manifests changes in ridge pattern brightness. Additionally, when $w_{app} = 0$, the generation exhibits no control over the appearance, having images generated with the same appearance latent look completely different. Figures 3b, 3c, 3d demonstrate the gradual appearance control improvement when the w_{app} is increased: from having small similarities when $w_{app} = 1$ (*e.g.*, some correlation between images in the second row of Figure 3b) to having almost identical appearances when $w_{app} = 20$ (*e.g.*, note the similar white triangle on the right corner of each image in the second row of Figure 3d). Furthermore, Figure 3 shows that setting higher w_{app} values increases the appearance variability for the same identity. As an example, for $w_{app} = 1$, ID3 has a similar appearance for all three appearance latent, while for every ID of $w_{app} = 5$ and $w_{app} = 20$ the appearances changes drastically for different appearance latent vectors.

4.1.2 Intra-class distribution of generated images

Next, we demonstrate the impact of w_{app} on the intra-class variability of fingerprints generated by FPGAN-Control. To measure the intra-class distribution, for each FPGAN-Control version, we randomly generated 1,000 synthetic identities. Each identity consists of two images generated using different appearance latents. For each pair of images, we measured two distances: cosine distance between the embedding vectors of the two images (using a ResNet18 recognition model) and their appearance distance computed by equation 4. Table 2 summarizes the results and verifies the trends observed in the qualitative evaluation. As

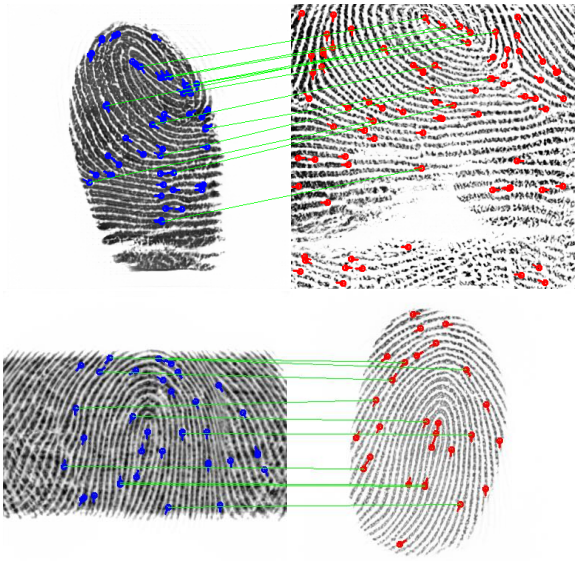


Figure 4. **Minutiae matching quality.** Examples of minutiae matching on genuine pairs (same identity per row) of our synthetic fingerprints when $w_{app} = 20$. The locations of minutiae points are annotated by circles and the orientations are indicated by the tails appended to each circle. The minutiae matcher then aims to find as many corresponding minutiae points as possible across the pairs. We note that even after placing heavy weight on our appearance loss, the minutiae defined identities are maintained across the image pairs as indicated by a large number of correspondences established.

the w_{app} increases the appearance distance calculated between images of the same identity grows, implying that the variance in appearance of same identity fingerprints is increasing.

4.1.3 Appearance control

We quantify the ability to control the appearance of FPGAN-Control. For each FPGAN-Control version, we randomly sampled 1,000 pairs of images generated with shared appearance latent and different identity latent. We computed the appearance distances, d_{app}, between all the pairs and report the results in Table 3. As expected, FPGAN-Control models trained with larger w_{app} exhibit better control over appearance. These results support the qualitative results presented in Figure 3, demonstrating the high level of appearance control achieved by FPGAN-Conrtol, where two images with different identity have nearly identical appearances.

4.1.4 Minutiae-points statistic

Since the early days when fingerprints first began to be studied, minutiae-points (Figure 4) have been a primary feature used to distinguish one fingerprint from another [18]. It is

Training dataset	Res18	Res34	Res50	Res101	Mob-050	Mob-100	Eff-s
Real data	90.54	89.72	92.00	93.69	93.74	94.22	93.69
StyleGAN2	23.54	6.085	6.47	8.87	30.70	32.37	5.59
PrintsGAN*	69.61	63.09	71.65	72.26	73.99	79.09	67.83
FPGC-0	87.53	87.25	87.01	87.60	88.72	90.13	83.87
FPGC-0.25	86.13	85.83	87.74	89.01	88.13	90.75	80.40
FPGC-0.5	88.00	87.24	86.71	87.26	88.20	90.50	80.72
FPGC-1	89.42	87.70	88.60	89.51	**90.63**	**91.14**	85.20
FPGC-5	87.48	86.97	87.81	90.15	88.72	89.68	82.82
FPGC-20	**89.57**	**89.72**	**90.19**	**91.08**	89.99	90.88	**88.55**
FPGC-0.25 + FPGC-20	91.60	91.59	91.99	90.70	92.53	93.22	88.96
FPGC-0.5 + FPGC-20	**92.15**	**92.24**	92.47	91.36	**92.62**	92.78	**90.60**
FPGC-1 + FPGC-20	91.86	91.88	**92.58**	87.33	91.98	**93.23**	89.88
FPGC-5 + FPGC-20	91.22	91.06	91.88	**91.69**	91.51	92.21	89.76

Table 4. **Recognition results for 50K synthetic identities.** TAR@FAR=0.1% results obtained by recognition models with various backbones trained using different synthetic datasets for the case of 50K synthetic identities. The datasets that were generated by FPGAN-control are denoted by FPGC-w_{app} where w_{app} corresponds to the weight of the appearance loss. We use an underline to denote models that surpass or are equal to the performance of models trained on real data.

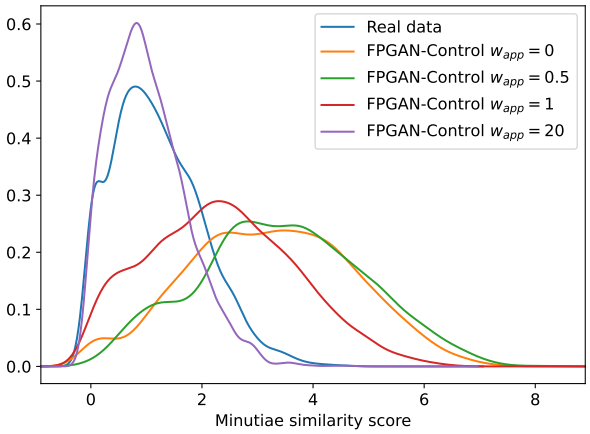


Figure 5. **Minutiae similarity score distributions.**

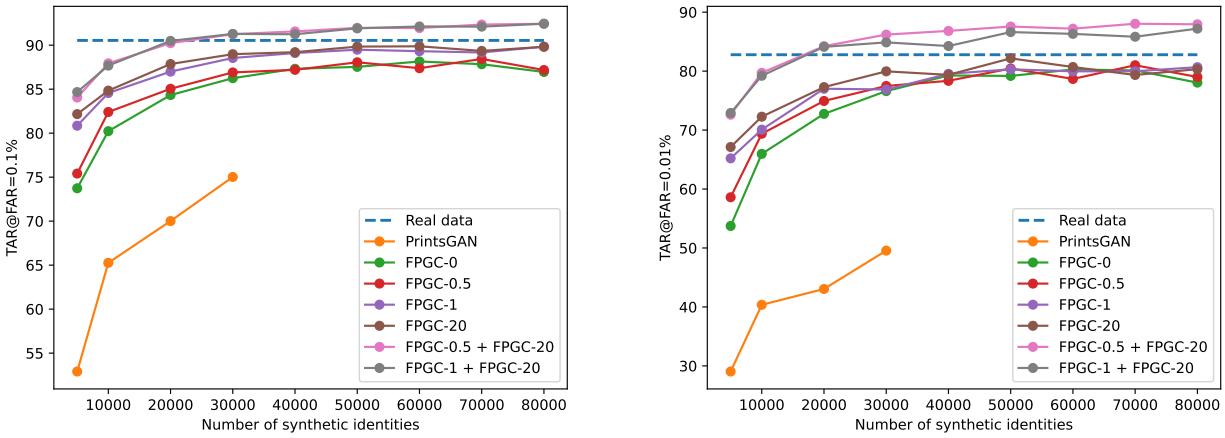
only in recent years that deep networks have been used to extract discriminative embeddings from fingerprints [12]. There are several benefits of using deep networks instead of minutiae matching approaches: (1) they allow faster matching[1] [12], (2) enable matching in the encrypted domain [14] and (3) can still perform successful matches when the fingerprint quality is very low [12]. To further show our ability to control the identity of fingerprints (as defined by minutiae points) in the presence of various impression styles (appearances), we computed the minutiae similarity score distributions of our synthetic fingerprints. We used the open source minutiae matcher from [8]. Figure 5 shows that as our appearance loss weight w_{app} increases from $w_{app} = 0$ to $w_{app} = 20$, the minutiae similarity score distributions shift towards the distribution of minutiae scores computed from real fingerprints. This lends additional strong evidence to our ability to maintain the identity of fingerprints as we modulate through different appearances.

[1]Minutiae matching approaches require expensive graph matching techniques.

4.2. Training with synthetic data

In this section we report the accuracy results obtained by recognition models trained using synthetic images that were generated by FPGAN-Control. We evaluated the performance of the trained recognition models on the test subset of the N2N dataset (real data). For training, we assumed that the real training dataset is no longer available and only synthetic identities were involved in the training process. To generate a synthetic identity, we have randomly sampled one common ID latent vector, $\mathbf{z}^{id}$, and 11 different appearance latent vectors, $\{\mathbf{z}_i^{app}\}_{i=1}^{11}$. We then concatenated the ID latent vector to each one of the appearance latent vectors, *i.e.* $[\mathbf{z}^{id}, \mathbf{z}_i^{app}]$. Finally, we provide the concatenated vectors as input to FPGAN-Control, generating 11 fingerprint images of the same identity, each having a different appearance. A synthetic dataset is constructed by generating multiple such synthetic identities.

In Table 4, we present the accuracy results (measured by TAR@FAR=0.1%) of various recognition models trained on different synthetic datasets using different network architectures. Specifically, we compare the results obtained by our FPGAN-Control to two baseline synthetic datasets. In the first, a regular StyleGAN2 [24] model was used to generate a multitude of synthetic fingerprint images. Then, each individual image was duplicated 11 times to define a



(a) ResNet18 TAR@FAR $= 0.1\%$.

(b) ResNet18 TAR@FAR $= 0.01\%$.

Figure 6. **Accuracy vs. number of synthetic identities used during training.** Real data corresponds to training the model with the entire 1,600 identities real dataset only, while the rest of the models were trained purely on synthetic identities. Note that PrintsGAN published only 35K identities, all of which were used in this evaluation.

unique identity. The second baseline is the publicly available dataset created by the PrintsGAN approach [13]. Multiple datasets were generated by the FPGAN-Control approach, each of which was named after the weight of its appearance loss. For example, FPGC-5 dataset was generated by setting $w_{app} = 5$ during the training of FPGAN-Control. We also combined images generated by multiple FPGAN-Control models to increase the diversity of the synthetic dataset.

From Table 4, we first observe a significant improvement of the proposed FPGAN-Control approach compared to the baseline datasets generated by the StyleGAN2 model and the PrintsGAN approach. Specifically, models trained using StyleGAN2's data struggled to converge, and training with PrintsGAN's data yielded poor recognition results. The significant leap in recognition accuracy obtained by FPGAN-Control demonstrates the effectiveness of the proposed approach in generating reliable fingerprint images that are useful for the task of training recognition models.

Secondly, in most cases, setting higher weight for the appearance loss of the FPGAN-Control model results in higher accuracy. For example, training with FPGC-20 is superior for the ResNet backbone family. In some cases, training a model using FPGC-20 can even achieve similar results compared to training with the original real data (e.g. training with 50K synthetic identities using ResNet-34 backbone). This shows the importance of the disentanglement between identity and appearance information that enables an increase in the variability of the fingerprint images generated by the FPGAN-control model.

In Figure 6 we present the TAR@FAR=0.1% and TAR@FAR=0.01% while gradually increasing the number of synthetic identities used during the training of the recognition model from 5K to 80K. We show that, as we increase the number of identities, the performance increases until it either plateaus or begins to deteriorate. We also report the results obtained by recognition models trained using PrintsGAN. The recognition accuracies obtained by models trained on data generated with PrintsGAN are inferior compared to any of the recognition models trained on data generated with FPGAN-Control. This might be partially due to the lack of sufficient appearance variation generated by PrintsGAN. Note that, for PrintsGAN, we evaluate the models using up to 30K synthetic identities, which constitute all the publicly available data released by the authors.

By combining multiple datasets generated by different FPGAN-Control models we obtain significant improvement of the recognition accuracy. For example, when using 80K synthetic identities, ResNet18 is able to achieve TAR@FAR=0.1% = 92.43%, an improvement of 1.61% compared to model trained by the real data. Incorporating images generated by various FPGAN-Control models further increases the variability of the images used for training, which in turn leads to better accuracy results.

5. Conclusions

We presented FPGAN-Control, a novel framework for training fingerprint generation models which can synthesize multiple images of the same novel fingerprint identity while controlling its appearance. We introduced a novel appearance loss for disentangling FPGAN-Control's latent space enabling control over generated fingerprint appearance while preserving their identity. Finally, the datasets generated by FPGAN-Control were used to train recognition models relying solely on synthetic identities and we showed that we are able to reach comparable and even higher accuracies than models trained using real data only.

References

[1] Afzalul Haque Ansari. Generation and storage of large synthetic fingerprint database. *ME Thesis, Jul*, 2011. 2

[2] Mohamed Attia, MennattAllah H Attia, Julie Iskander, Khaled Saleh, Darius Nahavandi, Ahmed Abobakr, Mohammed Hossny, and Saeid Nahavandi. Fingerprint synthesis via latent space representation. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1855–1861. IEEE, 2019. 2

[3] Gwangbin Bae, Martin de La Gorce, Tadas Baltrušaitis, Charlie Hewitt, Dong Chen, Julien Valentin, Roberto Cipolla, and Jingjing Shen. Digiface-1m: 1 million digital face images for face recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3526–3535, 2023. 1, 3

[4] Keivan Bahmani, Richard Plesh, Peter Johnson, Stephanie Schuckers, and Timothy Swyka. High fidelity fingerprint generation: Quality, uniqueness, and privacy. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3018–3022. IEEE, 2021. 1, 2

[5] Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9. IEEE, 2018. 2

[6] Fadi Boutros, Marco Huber, Patrick Siebke, Tim Rieber, and Naser Damer. Sface: Privacy-friendly and accurate face recognition using synthetic data. In *2022 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–11. IEEE, 2022. 1, 3

[7] Kai Cao and Anil Jain. Fingerprint synthesis: Evaluating fingerprint search at scale. In *2018 International Conference on Biometrics (ICB)*, pages 31–38. IEEE, 2018. 1

[8] Kai Cao, Dinh-Luan Nguyen, Cori Tymoszek, and Anil K Jain. End-to-end latent fingerprint search. *IEEE Transactions on Information Forensics and Security*, 15:880–894, 2019. 7

[9] Raffaele Cappelli, Dario Maio, and Davide Maltoni. Synthetic fingerprint-database generation. In *2002 International Conference on Pattern Recognition*, volume 3, pages 744–747. IEEE, 2002. 2

[10] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5154–5163, 2020. 1, 3

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1

[12] Joshua James Engelsma, Kai Cao, and Anil K Jain. Learning a fixed-length fingerprint representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 7

[13] Joshua James Engelsma, Steven Grosz, and Anil K Jain. Printsgan: Synthetic fingerprint generator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6111–6124, 2022. 1, 2, 8

[14] Joshua J. Engelsma, Anil K. Jain, and Vishnu Naresh Boddeti. Hers: Homomorphically encrypted representation search. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2022. 7

[15] Masud An-Nur Islam Fahim and Ho Yub Jung. A lightweight gan network for large scale fingerprint generation. *IEEE Access*, 8:92918–92928, 2020. 2

[16] César Augusto Fontanillo López and Abdullah Elbi. On synthetic data: a brief introduction for data protection law dummies. 2022. 1

[17] J. D. Grantham K. Ko K. Marshall M. Schwarz E. Tabassi B. Woodgate C. Boehnen G. P. Fiumara, P. A. Flanagan. Nist special database 302: Nail to nail fingerprint challenge, 2019. 4

[18] Francis Galton. *Finger prints*. Macmillan and Company, 1892. 6

[19] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014. 2

[20] Steven A Grosz and Anil K Jain. Spoofgan: Synthetic fingerprint spoof images. *IEEE Transactions on Information Forensics and Security*, 18:730–743, 2022. 2

[21] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[22] Peter Johnson, Fang Hua, and Stephanie Schuckers. Texture modeling for synthetic fingerprint generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 154–159, 2013. 2

[23] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 1

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1, 3, 7

[25] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[26] Adam Kortylewski, Andreas Schneider, Thomas Gerig, Bernhard Egger, Andreas Morel-Forster, and Thomas Vetter. Training deep face recognition systems with synthetic data. *arXiv preprint arXiv:1802.05891*, 2018. 3

[27] Marek Kowalski, Stephan J. Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton.

CONFIG: Controllable Neural Face Image Generation. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[28] Mandi Luo, Jie Cao, Xin Ma, Xiaoyu Zhang, and Ran He. Fa-gan: Face augmentation gan for deformation-invariant face recognition. *IEEE Transactions on Information Forensics and Security*, 16:2341–2355, 2021. 3

[29] Shervin Minaee and Amirali Abdolrashidi. Finger-gan: Generating realistic fingerprint images using connectivity imposed gan. *arXiv preprint arXiv:1812.10482*, 2018. 2

[30] Vishesh Mistry, Joshua J Engelsma, and Anil K Jain. Fingerprint synthesis: Search with 100 million prints. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2020. 1

[31] Jannis Priesnitz, Christian Rathgeb, Nicolas Buchmann, and Christoph Busch. Syncolfinger: Synthetic contactless fingerprint generator. *Pattern Recognition Letters*, 157:127–134, 2022. 2

[32] Haibo Qiu, Baosheng Yu, Dihong Gong, Zhifeng Li, Wei Liu, and Dacheng Tao. Synface: Face recognition with synthetic data, 2021. 1, 3

[33] M Sadegh Riazi, Seyed M Chavoshian, and Farinaz Koushanfar. Synfi: Automatic synthetic fingerprint generation. *arXiv preprint arXiv:2002.08900*, 2020. 2

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1

[35] Ataher Sams, Homaira Huda Shomee, and SM Mahbubur Rahman. Hq-fingan: High-quality synthetic fingerprint generation using gans. *Circuits, Systems, and Signal Processing*, 41(11):6354–6369, 2022. 2

[36] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4510–4520. Computer Vision Foundation / IEEE Computer Society, 2018. 5

[37] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 1

[38] Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gerard Medioni. Gan-control: Explicitly controllable gans. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14083–14093, 2021. 1, 2, 3

[39] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10096–10106. PMLR, 2021. 5

[40] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Perez, Michael Zollhofer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3D Control Over Portrait Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[41] H. Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jin Zhou, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 5

[42] André Brasil Vieira Wyzykowski and Anil K Jain. Synthetic latent fingerprint generator. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 971–980, 2023. 2

[43] André Brasil Vieira Wyzykowski, Mauricio Pamplona Segundo, and Rubisley de Paula Lemes. Level three synthetic fingerprint generation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9250–9257. IEEE, 2021. 1, 2

[44] Ziming Yang, Jian Liang, Chaoyou Fu, Mandi Luo, and Xiao-Yu Zhang. Heterogeneous face recognition via face synthesis with identity-attribute disentanglement. *IEEE Transactions on Information Forensics and Security*, 17:1344–1358, 2022. 3

[45] Qijun Zhao, Anil K Jain, Nicholas G Paulter, and Melissa Taylor. Fingerprint image synthesis based on statistical feature models. In *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 23–30. IEEE, 2012. 2