To appear in ICB 2018

Fingerprint Synthesis: Evaluating Fingerprint Search at Scale

Kai Cao and Anil K. Jain
Michigan State University, East Lansing, Michigan, USA
{kaicao, jain}@cse.msu.edu

Abstract

A database of a large number of fingerprint images is highly desired for designing and evaluating large scale fingerprint search algorithms. Compared to collecting a large number of real fingerprints, which is very costly in terms of time, effort and expense, and also involves stringent privacy issues, synthetic fingerprints can be generated at low cost and does not have any privacy issues to deal with. However, it is essential to show that the characteristics and appearance of real and synthetic fingerprint images are sufficiently similar. We propose a Generative Adversarial Network (GAN) to generate 512×512 *rolled fingerprint images. Our generative model for rolled fingerprints is highly efficient (12ms/image) with characteristics of synthetic rolled prints close to real rolled images. Experimental results show that our model captures the properties of real rolled fingerprints in terms of (i) fingerprint image quality, (ii) distinctiveness and (iii) minutiae configuration. Our synthetic fingerprint images are more realistic than other approaches.*

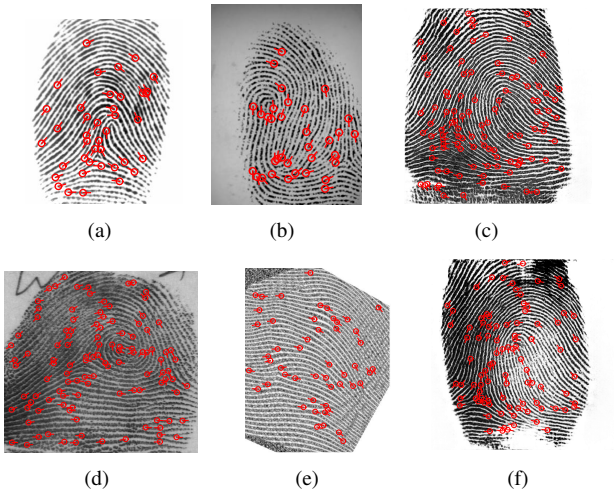

Figure 1. Comparing real fingerprint images ((a) and (d)) and synthesized fingerprint images ((b), (c), (e) and (f)). Images in (a) and (d) are from FVC2002 DB1 [13] consisting of plain impressions and NIST SD4 [3] consisting of rolled impressions, respectively. Images in (b) and (e) were synthesized using the approaches in [10] and [28], respectively, and images in (c) and (f) were synthesized by the proposed approach. All the above images are at a resolution of 500dpi.

1. Introduction

A fingerprint refers to the friction ridge patterns on a human fingertip [19]. Due to the perceived uniqueness and persistence of friction ridge patterns, large scale Automated Fingerprint Identification Systems (AFIS) have been widely deployed in both law enforcement and numerous civilian applications. For example, the FBI's Next Generation Identification (NGI) database, one of the world's largest law enforcement database, maintains about 120 million subjects[1], including both criminal and civil tenprints, for background check and criminal search. During August 2017 alone, 847,403 criminal searches and 3,397,167 civil searches were received by NGI[1]. Representative examples of large scale civilian applications include (i) the OBIM (formerly the US-VISIT) program by the Department of Homeland Security [4] and (ii) India's Aadhar project [5], which is now the largest biometrics deployment in the world with an enrollment that already exceeds 1.1 billion tenprints (along with corresponding irises and photos) of supposedly distinct individuals. For the task of identifying a fingerprint in such massive scale applications, high identification accuracies (low $FPIR$ and $FNIR$ values) and extremely efficient search are critical. As the sizes of these databases will continue to grow over time, there is a need to predict the scalability of current search algorithms

A number of fingerprint indexing[2] algorithms [23, 9, 8] have been proposed in literature to improve the search speed

[1] https://www.fbi.gov/file-repository/ngi-monthly-fact-sheet/view

[2] Biometric *search* examines biometric database against a probe to return either a candidate list or a comparison decision that the probe does or does not match with one or more references [ISO/IEC 2382-37:2017]. *Indexing* refers to a technique which speeds up the search by using special organization of data.

Table 1. A summary of major studies on fingerprint synthesis reported in the literature.

Algorithm	Orientation model	Minutiae model	Ridge structure model	Comments
Cappelli et al. [10]	Zero-pole model	random model	Gabor filtering	The three models are used independently; synthetic fingerprints do not visually appear to be realistic; no analysis on "fingerprintness" [27] of synthetic fingerprints; synthesizes plain fingerprints
Johnson et al. [17]	Zero-pole model	random model	Gabor filtering	The three models are used independently; synthetic fingerprints do not visually appear to be realistic; no analysis on "fingerprintness" of synthetic fingerprints; synthesizes plain fingerprints
Zhao et al. [28]	Zero-pole model	statistical model	AF-FM model	The three models are used independently; synthetic fingerprints do not visually appear to be realistic; no analysis on "fingerprintness" of synthetic fingerprints; synthesizes rolled fingerprints
Bontrager et al. [7]	Wasserstein GAN (WGAN)			128×128 fingerprint patches
Proposed	Improved WGAN using Autoencoder for initialization			Efficient algorithm for synthesizing 512×512 rolled fingerprint images. A database of 10 million synthetic rolled fingerprints was generated; Fingerprintness was analyzed and search experiments were conducted with two different galleries, one consisting of 250K real rolled prints and another with synthesized rolled prints.

and accuracy. However, there are only a few public domain fingerprint databases available to researchers for performance evaluation. To our knowledge, the largest publicly available fingerprint database is NIST SD14 [1] which has 54,000 fingerprint images from 27,000 fingers (two impressions per finger). Although some improvements in search algorithms' performance have been shown on this database [23, 9], it is not clear how these approaches will scale up with fingerprint database size, say 100 million fingerprints, comparable to those used in law enforcement and civil ID projects. It is well known that the identification accuracy drops almost linearly with the increase in gallery size [26]. In some of the studies [23, 8] proprietary fingerprint databases have been used to augment the gallery size for performance evaluation. For example, one million rolled fingerprints from various law enforcement agencies were used in [23] and 250,000 rolled fingerprints from a different law enforcement agency were used in [8]. However, these databases are still much smaller than operational databases.

In the absence of large public domain fingerprint databases, search accuracies in terms of $FPIR$ (False Positive Identification Rate) and $FNIR$ (False Negative Identification Rate) are estimated using FMR (False Match Rate) and $FNMR$ (False Non-Match Rate) values using

$$FNIR = FNMR, \tag{1}$$

$$FPIR = 1 - (1 - FMR)^N, \tag{2}$$

where N is the size of the database [16]. The limitations of using this formula are well known, see for example [12]. As sizes of the criminal and civil databases continue to grow, we need capabilities to accurately predict how the search accuracies will scale up using both theoretical analysis as well as empirical support. Furthermore, because some of the studies have used proprietary databases with possibly different characteristics, it is not possible to evaluate their search algorithms on a common database for a fair comparison. A large scale publicly available fingerprint database is therefore highly desired. Unfortunately, due to increased emphasis on privacy laws[3] related to fingerprints, it is unlikely that any government agency will release a large collection of fingerprints in the public domain. Over the last two years, NIST has taken down several public domain fingerprint databases (NIST SD27 [2] and NIST SD14 [1]) from its website that were available to researchers for many years. This was done apparently due to privacy issues.

Collecting a large number of real fingerprints is indeed costly in terms of time, effort and expense, and also involves stringent user consent and Institutional Review Board (IRB)[4] issues. The only option left is then to synthesize realistic fingerprint images [14, 28, 21]. The typical approach adopted in the synthesis process is as follows: (i) generate fingerprint ridge orientation field and minutiae using statistical models, (ii) use Gabor filtering [10] or the AM-FM model [28] to generate friction ridge pattern, and (iii) insert additive noise models [17] to make synthetic fin-

[3] http://www.sos.state.nm.us/uploads/files/CH36-HB15-2017.pdf

[4] https://en.wikipedia.org/wiki/Institutional_review_board

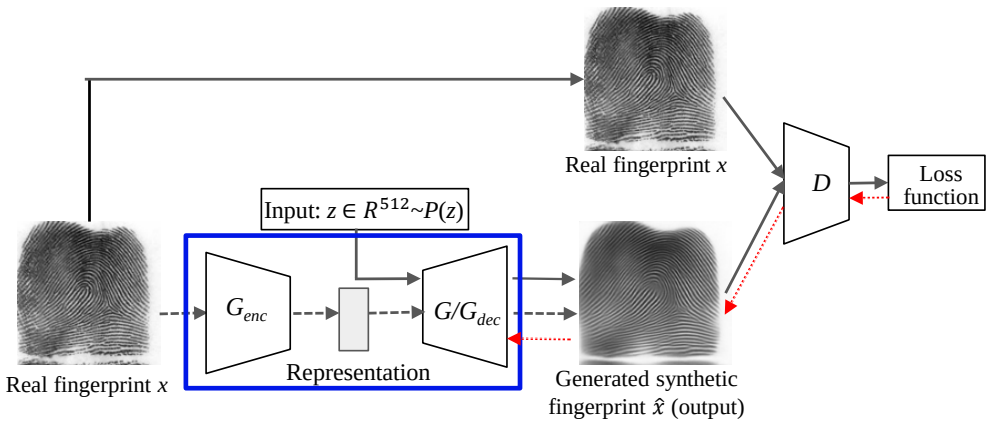

Figure 2. Flowchart of the proposed fingerprint synthetic generator. Dotted arrows show the training of the convolutional autoencoder, including encoder G_{enc} and decoder G_{dec}, and the solid arrows show the training of I-WGAN [15], including generator G and discriminator D, where the decoder G_{dec} is used to initialize fingerprint generator G. $P(z)$ is the 512-dimensional standard multivariate normal distribution. During fingerprint image synthesis, a 512-dimensional random vector is input to the generator G to obtain a synthetic fingerprint. The loss function, which a Wasserstein metric consisting of the probability distributions of real images and synthetic images and gradient penalty, is used to optimize the parameters in G and D using backpropagation (red dotted arrows).

gerprint images appear *realistic*. Table 1 summarizes some of the prominent published approaches to fingerprint synthesis. Fig. 1 compares real fingerprint images with synthesized fingerprint images for both plain and rolled fingerprints. There are a number of limitations of these published approaches:

1. The synthetic fingerprint images visually look different from the real fingerprint images, as can be seen in Figs. 1 (b) and (e), because of the additive noise to the master/ideal fingerprint images generated by Gabor filtering and AM-FM models. As a comparison, the corresponding synthetic images generated by our model are shown in Figs. 1 (c) and (f), respectively, which appear realistic.
2. Minutiae distribution models and minutiae sampling approaches do not consider the local minutiae configurations. Hence these approaches may generate invalid minutiae configurations. Gottschlich and Huckemann [14] showed that by using minutiae distribution alone, they could successfully classify real and synthetic fingerprints generated by SFinge [10]. Compare Figs. 1 (b) and (e).
3. Some unrealistic ridge patterns are generated due to the independent processes of minutiae sampling and ridge generation (compare Figs. 1 (b) and (e)).
4. A fixed ridge spacing (sum of the widths of ridge and valley) is typically used for generating a fingerprint image. By using two ridge spacing features with additional four features, Chen et al. [11] reported a 98% accuracy in separating synthetic fingerprints by SFinge [10] from real fingerprints.
5. Although all these approaches can generate any number of synthetic fingerprints, there was no validation done to determine the similarity between real and synthetic fingerprint images.

Deep Convolutional Generative Adversarial Networks (DCGAN) [22] have shown great promise in generating synthetic face images from random input vectors. Mai et al. [18] showed that by using the augmented face databases generated by DCGAN, the reconstructed face images from deep face templates can be used to successfully launch presentation attacks. Bontrager et al. [7] generated master prints (partial fingerprints) for presentation attacks using Wasserstein GAN (WGAN) [6], which leverages the Wasserstein distance to produce a value function with better theoretical properties than DCGAN. However, the sizes of synthetic image in these studies (128$\times$ 128 pixels in [7], and 96$\times$ 96 pixels in [18]) are significantly smaller compared to the typical size (512 $\times$ 512 pixels) of 500 dpi rolled fingerprints. Directly using these GANs to synthesize 512 $\times$ 512 rolled fingerprint images, in our experience, leads to synthetic fingerprint images that are similar to each other. In other words, there is no diversity among the set of 512x512 synthetic rolled fingerprints generated by DCGAN.

The contributions of this paper are as follows.

1. Synthesis of rolled fingerprint images (512$\times$512 pixels) using an improved-WGAN (I-WGAN) framework [15]. A Convolutional Autoencoder was trained for initializing the I-WGAN which improves synthetic fingerprint image characteristic and appearance.
2. Efficient synthesis procedure with an average time of 12ms per 512$\times$512 rolled fingerprint image. Thus we are able to generate 10 million synthetic rolled fingerprint images in a little over one day using a desktop with i7-6700K CPU@4.00GHz and GTX 1080 GPU.

3. Demonstrated the similarity between synthesized and operational rolled fingerprint images in terms of (i) fingerprint image quality, (ii) distinctiveness, (iii) minutiae configuration, and (iv) identification accuracies.

2. Proposed Fingerprint Synthesis

The proposed fingerprint synthesis is based on the improved-WGAN (I-WGAN) [15], which includes a generator (G) and a discriminator (D), also called a "critic" in [15, 6]. A convolutional auto-encoder (CAE), consisting of an encoder G_{enc} to extract a 512-dimensional representation from an input fingerprint and a decoder G_{dec} to reconstruct the input fingerprint from the representation, is trained. The decoder G_{dec} is then used to initialize the generator G to train the I-WGAN, which improves the fingerprint image quality and diversity. Fig. 2 shows the flowchart of the training process of the proposed fingerprint synthesis approach.

2.1. Improved-WGAN

Wasserstein GAN (WGAN) [6] leverages the Wasserstein distance to produce a value function which has better theoretical properties than the original DCGAN. However, WGAN requires that the discriminator or critic must lie within the space of 1-Lipschitz function. The authors of [15] showed that critic weight clipping can lead to pathological behavior and proposed an improved-WGAN. The generator (G) operates on a random vector input z, $z \sim p(z)$, to generate a fingerprint image $\hat{x}$ while the discriminator (D) outputs the critic value. The I-WGAN is trained to minimize the Wasserstein metric consisting of the probability distributions of real images and synthetic images and gradient penalty. The objective loss function (L) is defined as

$$L = \min_{G} \max_{D \in \Delta} E_{x \sim P_r}[D(x)] - E_{\hat{x} \sim P_g}[D(\hat{x})] + \lambda E_{\hat{x} \sim P_{\hat{x}}}[(\nabla_{\hat{x}} D(\hat{x}) - 1)^2], \quad (3)$$

where P_r is the real image distribution, P_g is the model distribution implicitly defined by $\hat{x} = G(z)$, $z \sim p(z)$, $P_{\hat{x}}$ denotes uniform sampling between pairs of points from the data distribution P_r and the generator distribution P_g, and Δ is the set of 1-Lipschitz functions. For details of I-WGAN, we refer the readers to [15].

Our I-WGAN architecture for 512×512 rolled fingerprint image synthesis, as shown in table 2, follows the guidelines suggested by Radford et al. [22]. There is one project and reshape layer and seven deconvolutional layers (also called fractional-stride [22]) in the generator G. The project and reshape layer generates 4×4 feature maps from a 512-dimensional input random vector. Each one of the seven convolutional layers has a kernel size of 4×4 and a stride size of 2×2 to successively enlarge the feature map size by a factor of 2. The output image is therefore of size 512×512. The numbers of channels output by the project and reshape layer and the subsequent 7 convolutional layers are 1,024, 512, 256, 128, 64, 32, 16 and 3, respectively. The discriminator D is essentially an inverse of G, except that the output of D is a scalar value in the range $(-\infty, \infty)$. ReLU activation is used in G for all layers except for the output, which uses tanh. LeakyReLU activation is used in D for all layers except for the output, which also uses tanh. Batch normalization, which stabilizes the learning process, is employed in both G and D.

2.2. Convolutional Autoencoder

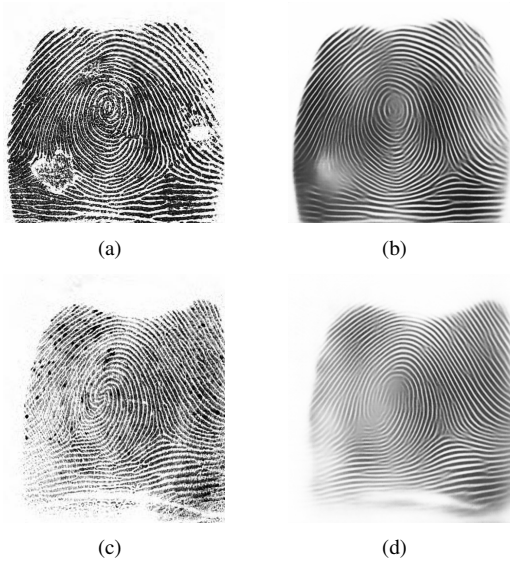

Figure 3. Examples of the reconstructed fingerprint images by the proposed CAE shown in Fig. 2. (a) and (c) 512×512 input rolled fingerprint images; (b) and (d) reconstructed 512×512 fingerprint images of (a) and (c), respectively.

Convolutional Autoencoder (CAE) models [20] are trained in an unsupervised mode to extract compact representations from unlabeled input. The encoder G_{enc} transforms an input into a low-dimensional representation, and the decoder G_{dec} is tuned to reconstruct the initial input from this representation through the minimization of a cost function. The architecture of the decoder G_{dec} is exactly the same as that of the generator G, so that G_{dec} trained by the CAE can be used to initialize G. The difference between D and G_{dec} is the last layer, where the output of D is a scalar value while the output of G_{enc} is a 512-dimensional feature vector. Let X be the set of training fingerprint images. The CAE is trained to minimize the objective L_{CAE}, which is defined as:

$$L_{CAE} = \sum_{x \in X} ||x - \hat{x}||_2^2, \quad (4)$$

where $\hat{x}$ is the reconstruction of $x \in X$ computed as $G_{dec}(G_{enc}(x))$. As shown in Fig. 3, compared to the in-

Table 2. The network architecture of I-WGAN generator. The values in Size In and Size Out columns follow the format of $height \times width \times \#channels$. The values in Kernel column follow the format of $height \times width$, $stride$. The architecture of the discriminator is the reverse (switch input and output) of the generator architecture and the output of discriminator is a scalar value activated by a LeakyReLU function.

Layer	Size In	Size Out	Kernel
Project and reshape	512×1 (1)	4×4 ×1024	-
conv1	4×4 ×1024	8×8 ×512	4×4, 2
conv2	8×8 ×512	16×16 ×256	4×4, 2
conv3	16×16 ×256	32×32 ×128	4×4, 2
conv4	32×32 ×128	64×64 ×64	4×4, 2
conv5	64×64 ×64	128×128 ×32	4×4, 2
conv6	128×128 ×32	256×256 ×16	4×4, 2
conv7	256×256 ×16	512×512×3	4×4, 2

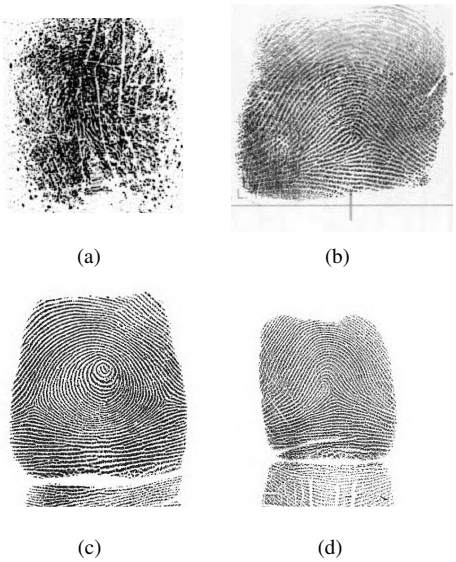

Figure 4. Examples of rolled fingerprint images used to train the generator for fingerprint synthesis. The NFIQ 2.0 [24] values of images in (a), (b), (c) and (d) are 1, 41, 62 and 93, respectively. Note that the range of NFIQ 2.0 value is from 0 to 100, with 0 being the lowest quality and 100 being the highest quality.

put fingerprint images, their reconstructed fingerprint images are of higher quality in terms of ridge clarity and contrast. Using G_{dec} to initialize I-WGAN generator thus ensures that it generates high quality fingerprint images.

2.3. Training Details

In order to train the CAE and I-WGAN, a rolled fingerprint database with 250K fingerprint images from a law enforcement agency [8] is used. Some example images of different quality from the training database are shown in Fig. 4. Although all these fingerprint images have the same resolution of 500 dpi, their sizes vary from 407×346 pixels to $750\times$ 800 pixels due to different sizes of friction ridge areas in them. Since 500 dpi fingerprint images of size $512\times$ 512 pixels cover sufficient friction ridge area for comparison, all the training fingerprint images are pre-registered using the approach in [8] and converted to $512\times$ 512 pixels. This pre-registration consists of the following steps: (i) orientation field estimation, (ii) reference point localization based on the orientation field, (iii) transforming orientation field against a dictionary with 36 orientation fields using the reference point based Hough transform, and (iv) image registration based on estimated orientation field transformation.

No data augmentation and image pre-processing is applied to the training set except normalizing the pixel values to the range [-1,1]. The model is implemented using tensorpack[5] and trained on a desktop with i7-6700K CPU@4.00GHz and GTX 1080 GPU. The network weights were initialized using a normal distribution with zero mean and a standard deviation of 0.02 and the batch size is set to 32. Adam is used for optimization, with a fixed learning rate of 2×10^{-4}, and a momentum of 0.5.

3. Experimental Results

The generation of synthesized fingerprint images is efficient. It takes about 12ms per image on a desktop with i7-6700K CPU@4.00GHz and GTX 1080 GPU. Thus, we were able to generate 10 million fingerprint images in 33 hours. In order to evaluate fingerprintness of the output images of our synthesis algorithm and compare it with real fingerprint images, we utilized the following 5 databases, where each dataset consists of 2,000 fingerprint images from 2,000 different fingers.

DB1 *NIST SD4*: First 2,000 images (f0001-f2000) from NIST SD4 [3].
DB2 *Proposed synthesis*: 2,000 synthesized images by the proposed approach.
DB3 *CASIA*: CASIA-FingerprintV5 database[6] consists of images from 4,000 fingers (2,000 fingers from each of right and left hands). We use the first impression of 2,000 fingers of the right hands.
DB4 *IBG Novetta*: 2,000 synthesized images by the IBG Novetta software [21].
DB5 *SFinge*: 2,000 synthesized images by SFinge [10].

Note that DB1 and DB2 are rolled fingerprint databases while DB3-DB5 are plain fingerprint image databases. Instead of comparing our synthesized rolled fingerprints to the synthesized plain fingerprints from [21] and [10], we are going to show that our synthesized rolled prints are closer to

[5]https://github.com/ppwwyyxx/tensorpack
[6]http://www.idealtest.org/dbDetailForUser.do?id=7

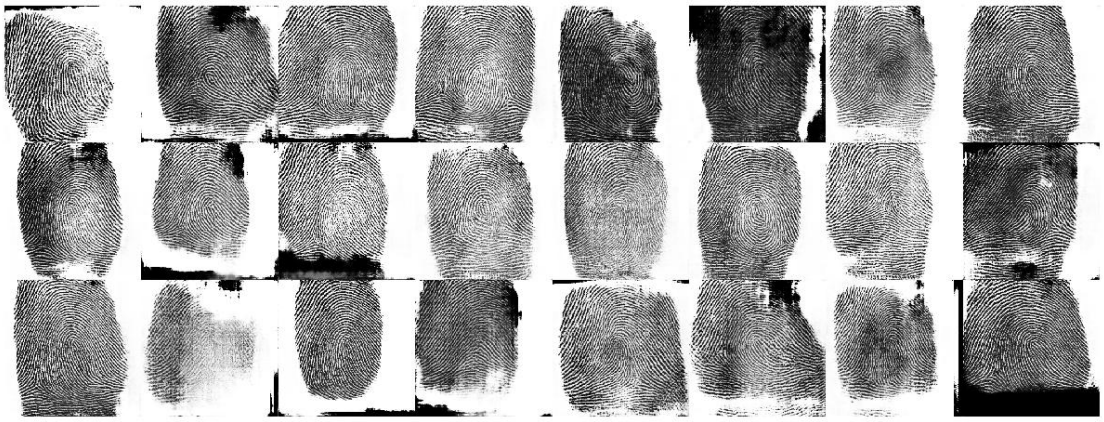

Figure 5. Examples of synthesized rolled fingerprint images (512×512 at a resolution of 500 dpi) by the proposed approach.

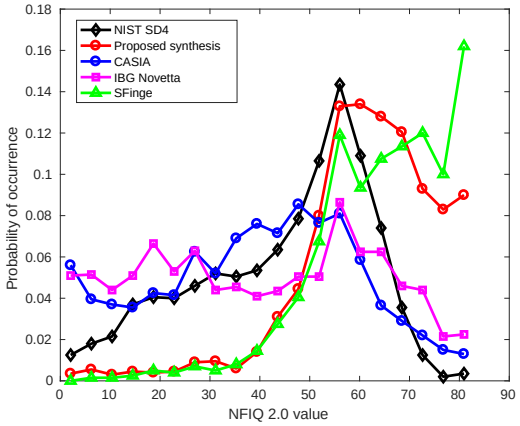

Figure 6. Distributions of NFIQ 2.0 values for images in the five databases. The range of NFIQ 2.0 value is [0,100] with 0 indicating the lowest quality and 100 being the highest quality value.

rolled prints in NIST SD4 than the synthesized plain fingerprints to the real plain fingerprints in CASIA database in terms of (i) fingerprint quality (Fig. 6), (ii) distinctiveness (Fig. 7) and (iii) distribution of minutiae configurations (Fig. 8).

3.1. Fingerprint Image Quality

NFIQ 2.0 (NIST Finger Image Quality) [24] is an open source software which is an updated version of the widely used NFIQ [25]. The range of output values of NFIQ 2.0 is [0, 100] with 0 being the lowest quality and 100 being the highest quality. Fig. 5 shows 24 randomly synthesized fingerprint images by the proposed approach. The distributions of NFIQ 2.0 values on the five datasets are shown in Fig. 6. The average NFIQ 2.0 values for the NIST SD4, proposed synthesis procedure, CASIA database, IBG Novetta and SFinge are approximately 44, 62, 39, 40 and 65, respectively. The proposed approach, like SFinge, tends to generate high quality fingerprint images because of our use of G_{dec} to initialize I-WGAN generator.

3.2. Distinctiveness

In order to evaluate the diversity of synthesized fingerprints, we compute the pair-wise comparison scores for each database using Verifinger SDK 6.3[7]. This leads to around 2 million $(2,000 \times 1,999/2)$ impostor scores for each database. Fig. 7 shows the five comparison score distributions in log scale. While these score distributions are very close each other, there are large variations at the higher range of score values. The maximum impostor comparison scores on DB1-DB5 are 51, 47, 110, 131 and 95, respectively. This indicates that the diversity of synthetic fingerprint images, in terms of their distinctiveness, is higher for the proposed approach than other two approaches [21, 10].

3.3. Minutiae Configuration

Fingerprint minutiae are regarded as the most discriminative features for fingerprint recognition [19]. Therefore, the spatial distribution of minutiae configurations extracted from the synthesized fingerprints is an indicator of their fingerprintness. Gottschlich and Huckemann [14] showed that 2D minutiae histogram (2DMH) is effective in separating real fingerprint images from synthetic fingerprints. In this section, we compare the 2D minutiae histograms to evaluate the fingerprintness of synthetic fingerprint images. Given an input fingerprint image, its minutiae set is extracted by Verifinger SDK 6.3, and its 2-dim. minutiae histogram is constructed by computing the pairwise minutiae distances ($d < 200$ pixels) in terms of location and directional difference (α in degrees). Specifically, the inter-minutiae distance d is divided into intervals of 20 pixels and inter-minutiae an-

[7] http://www.neurotechnology.com/verifinger.html

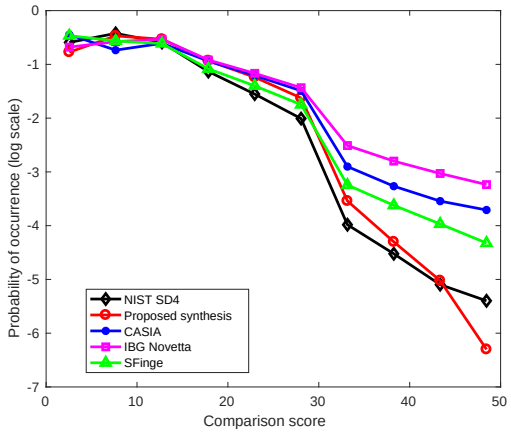

Figure 7. Impostor score distributions of the five fingerprint databases (DB1-DB5). The minimum and maximum comparison score values output by Verifinger SDK 6.3 are 0 and 6,356, respectively. Higher the score, more similar are the two fingerprint images. Note that the probability of occurrence is shown in log scale to illustrate the differences at the higher range of values.

gular difference α is divided into intervals of 18°. Hence the 2D minutiae histogram consists of 10×10 bins. Fig. 8 compares the average 2D minutiae histograms (Ave-2DMH) for the five fingerprint databases, DB1-DB5.

In order to compute the similarity between real fingerprint images and synthetic fingerprint images in terms of minutiae configurations, we use the Euclidean distances between the Ave-2DMHs. In order to alleviate the effect of fingerprint image types and number of minutiae, each Ave-2DMH is normalized with 0 mean and 1 standard deviation. Since DB1 and DB2 contain rolled fingerprint images, and DB3-DB5 contain plain fingerprint databases, we compute the inter-histogram Euclidean distances between DB1 and DB2, which is 2.96, and between DB3 and DB4 (DB5), which is 3.3 (3.94). This suggested that the synthesized rolled fingerprint images by our approach (DB2) are closer to real rolled fingerprint images (NIST SD4 in DB1) than the other two synthetic approaches whose images are in DB4 and DB5 [21, 10].

3.4. Fingerprint Search

Recall that one of main goals of this paper is to synthetically generate a large collection of fingerprints for large scale fingerprint search evaluation. Fingerprint search accuracy is also an indicator of the diversity of synthesized fingerprints, as shown in [26]; the identification accuracy drops with the increase in gallery size. In this section, we adopt a deep learning based fingerprint search algorithm proposed in [8] for the evaluation. To compare the search performance against 250K real rolled fingerprint images we used for training, we also generate 250K synthe-

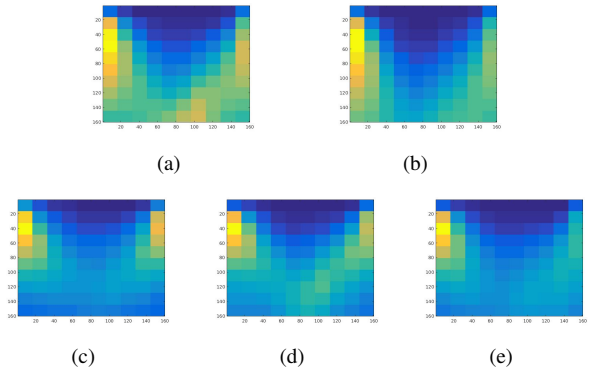

Figure 8. Images in (a)-(e) are average 2D minutiae histograms (shown as heat maps) of DB1(NIST SD4), DB2 (proposed synthesis approach), DB3 (CASIA), DB4 (IBG Novetta) and DB5 (SFinge), respectively. Recall, DB2, DB4 and DB5 contain synthesized fingerprint images. These images were upsampled by a factor of 16 for better visual quality.

sized fingerprint images. In addition to these two databases as gallery, two rolled fingerprint databases, namely, NIST SD4 (2,000 fingers, 2 impressions per finger) [3] and NIST SD14 (27,000 fingers, 2 impressions per finger) [1], are also used in the experiments. The "F" impressions in these two databases were used as gallery and "S" impressions were used as query. Fig. 9 compares the rank-20 identification accuracies against real fingerprint images and synthetic fingerprint images used to augment the gallery. The rank-20 identification accuracies against synthesized fingerprint images in the gallery on NIST SD4 (NIST SD14) are 98.7% (98.7%) to 96.1% (95.0%) for augmented gallery sizes 0 and 250K, respectively. This decrease shows that the downward trend in search accuracy w.r.t. background size is the same for synthesized rolled prints as real rolled prints. We do notice that the identification accuracies against synthetic fingerprints are about 5% higher compared to real fingerprints. This could possibly be explained due to the stochastic nature of our fingerprint generator. We plan to verify this by generating an ensemble of 250K rolled synthetic images.

4. Conclusions and Future Work

Generating synthetic fingerprints is a low cost and efficient way for designing and evaluating large scale fingerprint search algorithms. Moreover, it does not have any privacy issues unlike sharing real fingerprint images. However, available approaches for fingerprint synthesis generate intermediate features, namely, ridge orientation field, ridge frequency field and minutiae, separately, which can result in unrealistic fingerprint images. By leveraging the developments in Generative Adversarial Network (GAN), i.e., Improved-WGAN (I-WGAN), we are able to generate realistic quality 512×512 rolled fingerprint images. The

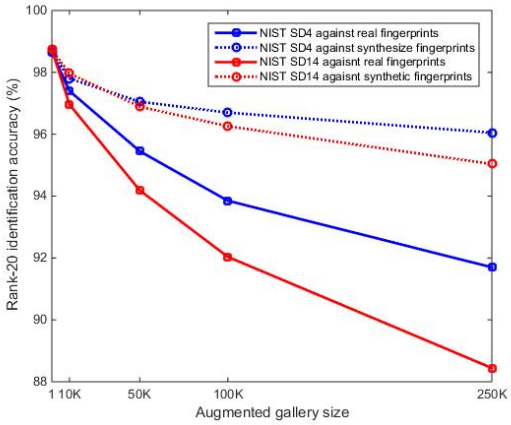

Figure 9. Rank-20 identification accuracies on NIST SD4 and NIST SD14 against real fingerprint images and synthesized fingerprint images with different augmented gallery sizes.

average time for generating a 512×512 rolled fingerprint image is about 12ms. Synthesizing rolled fingerprint images, not done previously, is essential for evaluating large-scale fingerprint search, particularly in law enforcement and forensic applications. Based on distributions of minutiae configurations and impostor scores by Verifinger SDK 6.3, the proposed synthetic fingerprints are closer to the corresponding distributions for real fingerprints. Future work includes (i) incorporating diversity criteria in training process and (ii) evaluating capacity of deep learning-based fingerprint recognition using GANs.

Acknowledgement

This research is supported by the NIST Forensic Research Program.

References

[1] NIST Special Database 14. http://www.nist.gov/srd/nistsd14.cfm.

[2] NIST Special Database 27. http://www.nist.gov/srd/nistsd27.cfm.

[3] NIST Special Database 4. http://www.nist.gov/srd/nistsd4.cfm.

[4] Office of Biometric Identity Management. http://bias.dhs.gov/obim.

[5] Unique Identification Authority of India. http://uidai.gov.in/.

[6] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. https://arxiv.org/abs/1701.07875, 2017.

[7] P. Bontrager, J. Togelius, and N. Memon. Deepmasterprint: Generating fingerprints for presentation attacks. https://arxiv.org/abs/1705.07386, 2017.

[8] K. Cao and A. K. Jain. Fingerprint indexing and matching: An integrated approach. In *IJCB*, 2017.

[9] R. Cappelli, M. Ferrara, and D. Maltoni. Fingerprint indexing based on minutia cylinder-code. *IEEE TPAMI*, 33(5):1051–1057, May 2011.

[10] R. Cappelli, D. Maio, and D. Maltoni. Synthetic fingerprint-database generation. In *International Conference on Pattern Recognition*, pages 744–747, 2002.

[11] S. Chen, S. Chang, Q. Huang, J. He, H. Wang, and Q. Huang. SVM-based synthetic fingerprint discrimination algorithm and quantitative optimization strategy. *PLOS ONE*, 9:1–9, 10 2014.

[12] J. Feng, A. K. Jain, and K. Nandakumar. Fingerprint matching. *Computer*, 43:36–44, 2010.

[13] FVC2002. http://bias.csr.unibo.it/fvc2002/. 2002.

[14] C. Gottschlich and S. Huckemann. Separating the real from the synthetic: minutiae histograms as fingerprints of fingerprints. *IET Biometrics*, 3(4):291–301, 2014.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of Wasserstein GANs. https://arxiv.org/abs/1704.00028, 2017.

[16] A. Jain, R. Bolle, and S. Pankanti. *Biometrics: Personal Identification in Networked Society*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[17] P. Johnson, F. Hua, and S. Schuckers. Texture modeling for synthetic fingerprint generation. In *IEEE Conference on CVPR Workshops*, pages 154–159, June 2013.

[18] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain. Face image reconstruction from deep templates. https://arxiv.org/abs/1703.00832, 2017.

[19] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Second Edition. Springer, 2009.

[20] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. *Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction*, pages 52–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[21] Novetta Biosynthetic Software. https://www.novetta.com/wp-content/uploads/2014/11/NOVBiosyntheticsOverview-2.pdf., 2014.

[22] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. https://arxiv.org/abs/1511.06434, 2015.

[23] Y. Su, J. Feng, and J. Zhou. Fingerprint indexing with pose constraint. *Pattern Recognition*, 54:1 – 13, 2016.

[24] E. Tabassi. NFIQ 2.0: NIST Fingerprint image quality. *NISTIR 8034*, 2016.

[25] E. Tabassi, C. Wilson, and C. Watson. Fingerprint image quality. *NISTIR 7151*, 2004.

[26] C. L. Wilson, C. I. Watson, M. D. Garris, and A. Hicklin. Studies of fingerprint matching using the NIST verification test bed (VTB). *NISTIR 7020*.

[27] S. Yoon and A. K. Jain. Is there a fingerprint pattern in the image? In *ICB*, pages 1–8, 2013.

[28] Q. Zhao, A. K. Jain, N. G. Paulter, and M. Taylor. Fingerprint image synthesis based on statistical feature models. In *IEEE BTAS*, pages 23–30, Sept 2012.