

Received April 9, 2020, accepted April 30, 2020, date of publication May 14, 2020, date of current version May 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994371

# A Lightweight GAN Network for Large Scale Fingerprint Generation

MASUD AN-NUR ISLAM FAHIM AND HO YUB JUNG<sup>ID</sup>

Department of Computer Engineering, Chosun University, Gwangju 61452, South Korea

Corresponding author: Ho Yub Jung (hoyub@chosun.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2017R1C1B5074302, and in part by the Research Fund from Chosun University, under Grant 2017.

**ABSTRACT** Generating fingerprint images for biometric purposes is both necessary and challenging. In this study, we presented a fingerprint generation approach based on generative adversarial network. To ensure GAN training stability, we have introduced conditional loss doping that allows a continuous flow of gradients. Our study utilizes a careful combination of a residual network and spectral normalization to generate fingerprints. The proposed average residual connection shows more immunity against vanishing gradients than a simple residual connection. Spectral normalization allows our network to enjoy reduced variance in weight generation, which further stabilizes the training. Proposed scheme uses spectral bounding only in the input and the fully connected layers. Our network synthesized fingerprints up to 256 by 256 in size. We used the multi-scale structural similarity (MS-SSIM) metric for measuring the diversity of the generated samples. Our model has achieved 0.23 MS-SSIM scores for the generated fingerprints. The MS-SSIM score indicates that the proposed scheme is more likely to produce more diverse images and less likely to face mode collapse.

**INDEX TERMS** Skip connection, spectral normalization, loss doping, mode collapse, diversity.

## I. INTRODUCTION

Fingerprints are essential to biometric systems. Many approaches have been proposed to create fingerprints for biometric purposes. Previous studies on fingerprint synthesis have been mostly associated with manual generation from a single or multiple fingerprint base structure. These studies show that using morphological or minutiae point manipulation, we can produce synthetic fingerprints [3]–[6].

A master fingerprint is capable of bypassing a small-scale biometric security system such as those used on smartphones. This MasterPrint invasion is possible due to the input data limitation of these devices. This scenario was introduced by [7], [8], who attempted to emulate the master fingerprint. Their study has shown that a master fingerprint can be obtained by manipulation of original images or through synthesis using the hill-climbing scheme. However, their master fingerprints are visually distinguishable from the original data. These studies [9]–[11] proposed a zero-pole model for synthetic fingerprint generation. However, visually indistinguishable fingerprints are still very difficult to

produce. Compared to previous studies, fingerprint generative adversarial network-based studies [12]–[14] have shown great promise. A summary of these studies is presented in table 1.

The GAN [1] is a massive leap in the area of artificial intelligence since it popularizes information synthesis using a neural network. The GAN framework takes a sampled data distribution and produces a synthetic data distribution that closely reflects the input data. Despite introducing an enormous number of possibilities, the GAN framework also has challenges, such as mode collapse, training stability, and a large computational budget. To mitigate these challenges, many schemes have been proposed, and all of them are based on the idea of the deep convolutional generative adversarial network (DCGAN) [15]. DCGAN uses deep convolutional layers instead of fully connected layers to generate fake images. It can successfully produce 64 by 64 images, somewhat successful with 128 by 128 size, but fails for larger scale.

Following DCGAN, three significant studies [16]–[18] were reported in 2017. Instead of typical cross-entropy-based loss, Arjovsky *et al.* [16] uses the Wasserstein distance as the loss function. This particular method enables the network to enjoy a continuous flow of loss values

The associate editor coordinating the review of this manuscript and approving it for publication was Xian Sun<sup>ID</sup>.

**TABLE 1. Previous fingerprint synthesis studies at a glance.**

Algorithm	Alignment model	Minutiae model	Ridge structure model	Approaches
Cappelli et al. [9]	Zero-pole model	Random model	Gabor filtering	Three models were used autonomously. However, the visual distinction between synthetic and real fingerprints is clear. No metric for measuring similarity between the generated fingerprints was provided.
Johnson et al. [10]	Zero-pole model	Random model	Gabor filtering	Three models were used autonomously. However, the visual distinction between synthetic and real fingerprints is clear. No metric for measuring similarity between the generated fingerprints was provided.
Zhao et al. [11]	Zero-pole model	Statistical model	AF-FM model	Three models were used autonomously. However, the visual distinction between synthetic and real fingerprints is clear. No metric for measuring similarity between the generated fingerprints was provided.
Bontrager et al. [13]	Wasserstein GAN (WGAN) equipped generator			128×128 slices
Kai et al. [12]	Auto encoder and decoder			Produces up to 512×512. Autoencoder-based image generation always approximates images close to the original image.
Shervin et al. [14]	DCGAN equipped generator			Produces 128 × 128 patches; produces blurry ridges. Additionally, like all other previous studies, they did not show any diversity analysis.

throughout the training in comparison to cross entropy-based loss functions. This method also improves Lipschitz constraint enforcement by proposing weight clipping. Later, Gulrajani *et al.* [17] improved weight clipping by introducing gradient penalty and it has been adopted by other GAN studies. An alternating gradient update-based scheme [18] has shown convergence when the distribution from the generator and the original distribution are absolutely continuous [19]. A much more stable training strategy was later introduced by Google [20] based on an encoder-decoder-based discriminator. This study employs the idea of the Wasserstein distance for constructing their adaptive learning rate dependent loss function. BEGAN [20] studied the training stability issue and successfully produced 128 by 128 images.

Currently, we have a handful of practical advances in GAN training mechanisms. However, the underlying dynamics have yet to be established. From [19], the authors of Dirac-GAN show that if it is possible to examine the eigenvalues of the Jacobian of the associated gradient vector field, local convergence and stability properties can be analyzed. A far-fetched similarity can be shown in the spectral analysis of the generative adversarial network [21]. This study place emphasis on normalizing the weights of the neural network to stabilize the training procedure. Similar to BEGAN [20], these studies [19], [21] examined the means of stable training to generate 128 by 128 images. Enforcing the Lipschitz continuity may lead to stable training [21]. This intuition is also supported by Odena *et al.* [22] since they have shown that the stability of the generator is dependent on taming the Jacobian. Another stable training scheme from Google [23] has also supported spectral analysis for the further stabilization of GAN training.

Instead of dealing with a cost function, some research suggests looking at the differences in the number of networks [24]–[27]. They proposed that multiple generators or discriminators can improve GAN stability. One study [24] shows that using the KL divergence and the reverse KL divergence as dual discriminators for GAN can improve training. These studies [25], [26] employ more than two discriminators to produce gradients with low variance, consequently improves the generator. However, this idea increases the training cost

dramatically for higher-dimensional images. AdaGAN [28] utilizes the boosting technique to improve training performance. This study [29] uses multiple generators to improve the GAN training scheme. Mixture GAN (MGAN) [30] proposes to reduce the mode collapse situation by training multiple generators to learn the statistics of different data modes. However, this increases training difficulty, and mode collapse remains unsolved.

The goal of this study is to design a GAN system to produce faithful synthetic fingerprint images. Synthetic fingerprints can aid in many research applications. For example, an extensive collection of usable synthetic fingerprint images can help in fingerprint detection, fingerprint classification, fingerprint liveness detection, or data augmentation for deep learning tasks.

Due to hardware, computational, and design complexities, it is hard to generate images with a higher dimension. We have designed the proposed network in a way that it can easily deal with the above concerns. To ensure stability in the fingerprint generation, we adopted spectral normalization [21]. The presence of skip connection in the generator and the discriminator helps our network to mitigate the vanishing gradient problem. We provided a more detailed analysis in the methodology section.

Here, we have proposed a GAN scheme that can successfully produce fingerprint patches. The overall contributions of this study can be summarized as follows:

- The proposed lightweight network can generate fingerprint patches for up to 256 by 256 fingerprint images.
- This study introduced loss-doping for overall training stability. Loss doping allows our network to avoid training collapse, which is prevalent in previous GAN studies. Also, we have observed improved convergence for this technique.
- Mode collapse is still an open challenge in GAN research. The proposed network is fairly free from this drawback, with an average MS-SSIM score of 0.23. We have observed that produced images are less likely to be similar to each other. Additionally, the utilization of data augmentation has enabled it to cope with the same image with different appearances.
- Our minimalistic residual-spectral network for fingerprint generation enjoys good stability during the entire

training procedure and is less likely to suffer from training collapse.

We have followed the usual sequence for documenting of our study. The very next section is focused on related works that produce higher resolution images. After this, we explain the network architecture and relevant analysis. Then, we present our result analysis, which is followed by the conclusion.

## II. RELATED WORK

The generation of high-resolution images has attracted much attention in recent years. Even though the DCGAN architecture enables the production of higher resolution images, it needs several modifications; it is unable to ensure fidelity and stability in image generation with higher dimensionality. These problems were later addressed in other studies, and many remedies have been proposed. StackGAN [31] and StackGAN++ [32] have tried to solve these problems gradually. In the earlier version [31], they used a two-stage GAN strategy to achieve higher resolution images. Using the attention mechanism, self-attention GAN [33] has been proposed.

ID-GAN [34] uses variational auto-encoder (VAE) to distill latent distribution for GAN training and produced high dimensional images with the help of 3 networks. RAGAN [34] produces high-quality images by decreasing the probability of fake data to be recognized as real for the generator. MSG-GAN [35] allows gradient flow between the generator and the discriminator, which results in 1k resolution images. This study [36] utilizes domain translation from semantic label maps to produce crisp HD cityscape images. AE-GAN [2] combines WGAN and VAE to create stable, high-resolution photos. COCO-GAN [37] generates state-of-the-art images by utilizing spatial information as the constraint for the generator. The semantic bottleneck network combines progressive semantic generation network and segmentation-to-image synthesis network to produce 5k images. BigGAN [23], Progressive GAN [38], StyleGAN [39] these state-of-the-art methods provide means for large scale ( $\geq 512$  by 512) image generation.

On the contrary, super-resolution GAN [40] can provide computational-friendly support to produce a high-resolution image. G-GANISR [41] improves the GAN performance for super-resolution by utilizing the least square loss function. Dual generative adversarial network [42] uses two generators to enhance the robustness of the network and successfully produces super-resolution images.

Very little has been done regarding studies on producing fingerprints compared to other GAN studies [13]. This study proposed Wasserstein distance-based GAN for fingerprint generation. Usually, GAN-based systems have advantages in producing sharper images than those produced by autoencoder-based schemes. Despite this trend, this scheme [13] seems to produce blurrier images. Additionally, compared to the original samples, the ridges presented in the images are more likely to be noisy. FingerGAN [14] proposed

a DCGAN-based scheme that emphasizes the TV loss as an extension to the traditional loss for the DCGAN. They have produced 512 by 512 images. Even with this large-scale synthesis, images seem to be fuzzy in comparison with the original images. The stability power of an autoencoder combined with WGAN [12] provides a good way to implement fingerprint generation compared to [13], [14]. They [12] have produced 512 by 512 images and have claimed to produce millions of samples in one day. In contrast, they have presented very little pictorial representations. Moreover, none of these studies [12]–[14] have presented a diversity analysis. We can summarize the remaining challenges in previous studies as follows:

1) Their schemes rely on Gabor filtering, and AM-FM modeling produces a visually different image that appears synthetic when juxtaposed with real fingerprints.

2) Minutiae-based modeling is independent of minutiae formation. This results in poor pattern generation for fingerprint synthesis.

3) Due to independent minutiae sampling, the generated ridges are very unrealistic.

4) Additionally, they cannot produce random realistic looking patterns, and the gaps between ridges are constant in the generated images [12]–[14].

In this study, we aim to produce fingerprint images that are free from the above traits. Additionally, we hope to produce faithful fingerprint images and least likely to face usual GAN training challenges.

## III. PROPOSED TRAINING SCHEME

This section will cover our reasoning regarding the proposed training scheme for fingerprint generation. A theoretical framework for spectral normalization-based training, training without spectral normalization, and the intuition behind the proposed loss function will form the bulk of this section.

In the adversarial training setup, we need a generator  $G$  and a discriminator  $D$ . The generator produces fake images to outperform the classification performance of the discriminator. Intuitively, we can consider a value function  $V$  for both of the networks to express the GAN formulation theory [1]. The original formulation of the generative adversarial network can be given by:

$$\min_G \max_D V(G, D) \quad (1)$$

Throughout the entire training, the generator  $G$  is trained to minimize its cost, while the discriminator  $D$  is trained to maximize its cost. For this, let  $x$  denote the sample data and  $z$  denote the noise data for the generator. Now,  $p_G$  is the distribution from the generator, and  $q_{data}$  is the distribution over  $x$ . Then, the conventional formulation of equation 1 can be stated as follows:

$$V(G, D) = E_{x \sim q_{data}} [\log(D(x))] + E_{z \sim p_G} [\log(1 - D(G(z)))] \quad (2)$$

In equation 2, we ensure that the discriminator is accurate over the real data by maximizing  $E_{x \sim q_{data}} [\log(D(x))]$ . On the

other hand, the generator  $G$  is producing data  $G(z)$  from the noise data  $z$ . The goal of the generator is to minimize  $E_{z \sim p_G}[\log(1 - D(G(z)))]$  by producing high quality fake data  $G(z)$ .

Let us define the training parameters for the discriminator and the generator as  $\Omega_D, \Omega_G$ . The discriminator maps the incoming data distribution in the network into a set of probabilities. These probabilities indicate which sample comes from the fake and real distributions. If  $P(x_i; \Omega_D)$  denotes the probability for the real data distribution that the discriminator classifies as real data and  $P(G(z_i); \Omega_D)$  denotes the probability for the generated distributions that the discriminator classifies as real data, then we can set up the cost function for our discriminator and generator as stated in equations 3 and 4.

$$\mathcal{L}_D = -\frac{1}{m} \sum_{i=1}^m [\log P(x_i; \Omega_D)] + [\log(1 - P(G(z_i); \Omega_D))] \quad (3)$$

$$\mathcal{L}_G = -\frac{1}{m} \sum_{i=1}^m [\log(1 - P(G(z_i); \Omega_D))] \quad (4)$$

Here,  $\mathcal{L}_D$  and  $\mathcal{L}_G$  denote the cost function for the discriminator  $D$  and the generator  $G$ , respectively. The data distribution for the discriminator is represented by  $x_i$ .  $G(z_i)$  represents the data distribution generated by the generator throughout the training, and  $z_i$  stands for the noise vector.

One important concept we should keep in mind is that the generator acts as a black box. Thus, weights generated for the generator are not independent of the discriminator. This means that if the discriminator is not strong enough to classify fake images, we will see that the generated distributions are poor in terms of fidelity. This concept brings about the idea of an optimal discriminator [16], [17], [21]. We can find out the theoretical representation for the optimal discriminator  $\mathcal{D}_G^*(x)$  by fixing the generator. If  $\mathcal{S}$  denotes the sigmoid function, then the optimal discriminator can be formulated as follows:

$$\mathcal{D}_G^*(x) = \frac{q_{data}(x)}{q_{data}(x) + p_G(x)} = \mathcal{S}(\lambda(x)) \quad (5)$$

Here,  $\lambda(x)$  is as follows:

$$\lambda(x) = \log_{q_{data}}(x) - \log_{p_G}(x) \quad (6)$$

Its derivative can be written as follows:

$$\delta_x \lambda(x) = \frac{1}{q_{data}(x)} (\delta_x q_{data}(x)) - \frac{1}{p_G(x)} (\delta_x p_G(x)) \quad (7)$$

The above equation is unbounded and may even be incomputable [21]. To mitigate this, some bounding conditions are necessary for convergence. This is the reason why researchers [16], [17], [21] have tried to bind the discriminator by K-Lipschitz, which is,

$$\max_{\|f\|_{lip} \leq k} V(G, D) \quad (8)$$

Here,  $\|f\|_{lip}$  denotes the smallest value  $y$ , where  $\|f(m) - f(m')\| / \|m - m'\| \leq y$  for any  $m, m'$ . This situation can be addressed by introducing spectral

normalization [21]. Spectral normalization stabilizes the training by normalizing the weights in any layer  $\xi$ . For a matrix  $A$ , the spectral norm  $\sigma(A)$  is defined as follows:

$$\sigma(A) = \max_{\xi \in \mathbb{R}^n, \xi \neq 0} \frac{\|A\xi\|_2}{\|\xi\|_2} \quad (9)$$

This equation resembles the maximum among the singular values of the matrix  $A$ . For a given vector  $s$  and weight matrix  $w$ ,

$$\sigma(w_\Omega, s) \leq \prod_{n=1}^N \sigma(w^n) \quad (10)$$

which means that to bound the spectral norm, its enough to normalize the spectral norm of  $w^n$  in each layer. The theoretical guarantee for the above equation can be obtained from [21].

From the fundamental assumptions of convex optimization, to ensure convexity, a multidimensional linear function has to be Lipschitz continuous. Spectral normalization controls the Lipschitz constant of the discriminator by constraining spectral norm in every single layer [21]. If the previous statement holds, then Lipschitz constant is the largest singular value of the linear function. In other words, it is the spectral norm. If any multidimensional linear function  $M$  is K-Lipschitz at 0, then it is K-Lipschitz at any other point. This property simplifies Lipschitz continuity as follows:

$$\|M\xi\| \leq K\|\xi\|; \quad \forall \xi \in I \quad (11)$$

Here,  $I$  is the distribution domain. We can write the above equation as follows:

$$\langle M\xi, M\xi \rangle \leq K^2 \langle \xi, \xi \rangle \quad (12)$$

This can be rewritten as follows:

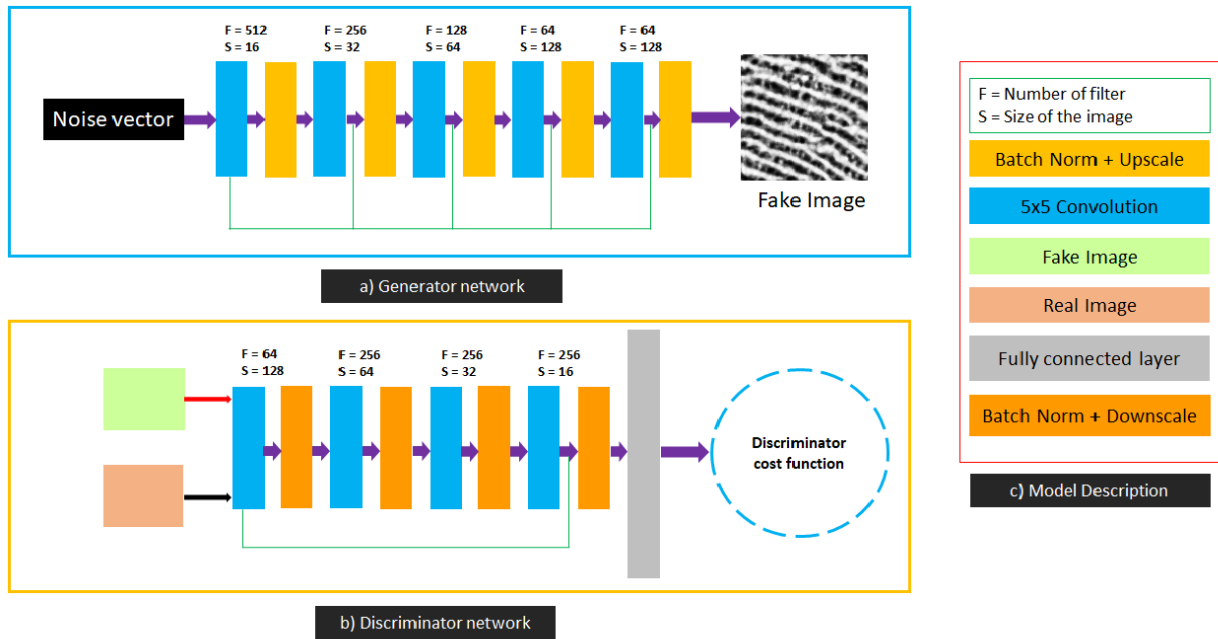
$$\langle (M^T M - k^2)\xi, \xi \rangle \leq 0 \quad (13)$$

Expanding  $\xi$  on the basis of eigenvector's orthonormality,

$$\begin{aligned} \langle (M^T M - k^2)\xi, \xi \rangle &= \langle (M^T M - k^2) \sum_i \xi_i v_i, \sum_j \xi_j v_j \rangle \\ &= \sum_i \sum_j \xi_i \xi_j \langle (M^T M - k^2) v_i, v_j \rangle \\ &\Rightarrow \sum_i (k^2 - \lambda_i) \xi_i^2 \geq 0 \end{aligned} \quad (14)$$

From the above equations,  $M^T M$  is positive semi-definite, which means all the values of  $\lambda_i$  must be non-negative. To ensure this, each of  $(k^2 - \lambda_i) \geq 0$ . Since the value of  $K$  is the minimum to satisfy the above constraint, then it is obvious from the above relationship that  $K$  is the square root of the largest eigenvalue of  $M^T M$ . Hence, the Lipschitz constant of any linear function is its spectral norm. This inherent property justifies the utilization of spectral normalization to ensure convergence [21]. We can only speculate that these properties also carry over to more complex non-linear models.

In our network setup, we used spectral bounding only in the dense and input layers of the discriminator. This makes



**FIGURE 1.** Proposed structure for the generator and the discriminator. a) This schematic diagram is indicating the details of the proposed generator. b) This schematic diagram is indicating the details of the proposed discriminator. c) This section is showing the meaning behind each block as presented in a and b.

our spectral norm-dependent setup different from [21], [23], where the authors have used it for every layer. We observed that this bounding has contributed to fingerprint generation by introducing more diversity. Our network achieved the best diversity score with the help of spectral normalization.

A vanishing gradient is a common challenge in GAN-based networks. To mitigate this, we can simply use residual connections. This residual formation has also been utilized by other studies [20], [21], [23] for this purpose. Let us say that the initial layer for our network is  $X^0$ . After applying the common activation function  $\Psi()$ , we can write this as  $\alpha^1 = \Psi(X^0)$ . The  $n$ th layer of the proposed generator is stated as follows:

$$\alpha^n = \Psi^n(X^{(n-1)}; \Omega^n) + \Psi^n(X^0; \Omega^1) = \alpha^{(n-1)} + \alpha^1 \quad (15)$$

We experimented with the skip connections using different algebraic operations. We found that simply adding the distributions from  $\alpha^1$  creates gradient explosion and convergence difficulty. We performed an averaging operation in skip connections instead of simple addition. This ensures the mitigation of the gradient-related problems that were typical of previous GANs. From figure 1, we used the skip connection in every layer for the generator. For the discriminator, we used it only in the last layer. We maintained this structure for training with and without spectral normalization.

For the objective function, we applied some of the modifications to obtain a better result. In GAN theory, there is no incentive for the GAN training scheme to reach a minimal point [19]. Even though it reaches a point where it can successfully produce high-quality images, it does not have the motivation to stop the training. Additionally, it can

propagate towards an unstable point. We found this through some experiments with the DCGAN. Similar findings were also observed by researchers from Google [23].

#### Training Scheme: Conditional Loss Doping

```

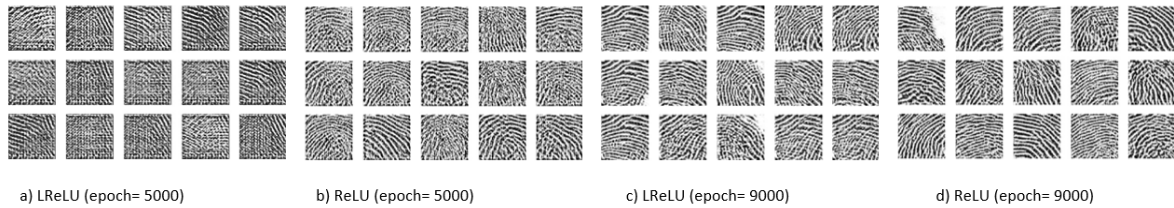
1  Data: Noise Vector, Real images
2  Output: Synthetic fingerprint images
1  For e=1 to k do,
2       $G'_{loss} = E_{x' \sim P_G}[\log(1 - D(x'))]$ 
3  If  $G'_{loss}[e] = G'_{loss}[e - 1]$ 
4       $G_{loss} = (1 + \beta) * G'_{loss}$ 
5  else  $G_{loss} = G'_{loss}$ 
6       $D_{loss} = E_{x \sim P_D}[\log(D(x))] + G_{loss}$ 
7  End
```

During the training procedure, we observed that a minute random modification in the generator loss can introduce dramatic changes. These changes rely on the degree of loss modification. This observation motivated us to take a different approach to mitigate the static loss scenario. In the traditional setup, when the discriminator wins over the generator, the discriminator and generator loss remain the same for some epochs. This static loss generation continues until the generator produces better images to fool the discriminator. To avoid this situation, we introduced loss doping. Loss doping implements a minute loss augmentation in the generator loss instead of returning its original loss value. We implement this only if the generator produces the same loss value for two consecutive epochs. The total procedure is presented in the training scheme above. In this way, our network has enjoyed a non-freezing epochs during the entire training time.



**TABLE 2.** Summary of loss augmentation in our network architecture.

Loss augmentation choice	Comments
Hinge Loss + Loss Augmentation	With spectral normalization, it works well at the beginning. However, later, it shows performance degradation
Huber Loss + Loss Augmentation	It fails to produce any sort of meaningful image structure
Binary Cross entropy + Total Variation	Severely unstable performance due to rapid total variation
Sigmoidal Cross entropy	Produces very poor-quality images but is better than the abovementioned cases

**FIGURE 2.** a) 128 by 128 patch generation using Leaky ReLU at epoch = 5000. b) 128 by 128 patch generation using ReLU at epoch = 5000. c) 128 by 128 patch generation using Leaky ReLU at epoch = 9000. d) 128 by 128 patch generation using ReLU at epoch = 9000. From this figure, sharpness and clarity achieved with ReLU activation is clearly visible.

Intuitively, loss doping acts more like a ‘conditional momentum’ scenario. Momentum allows the gradient descent algorithm to escape the saddle point and push the optimization procedure towards the convergence. In this study, loss doping does the same by changing the loss values minutely if and only if consecutive loss value is the same. This conditional doping supposedly changes the trainable weights and helps the optimization to converge faster. Empirically, we have observed faster convergence with the help of loss doping. Figure 10 and figure 11 show the effect of loss doping.

To ensure the best means of loss doping, we experimented with piecewise loss difference, random numbers, and percentile loss. With percentile loss augmentation, we observed stable training without facing training collapse. In our work, we updated the generator loss  $G_{\text{loss}}$  using loss doping as follows:

$$G_{\text{loss}} = (1 + \beta) * G_{\text{loss}} \quad (16)$$

Here,  $\beta$  is the doping amount for our network. We maintained the value of  $\beta = G_{\text{loss}}/10000$  for the training. Due to this, our network enjoyed a variable learning rate during the entire training time. In table 2, the summary of the loss doping is noted.

We did not limit our experiments only to networks for fingerprint generation. For maintaining stable training, we experimented with weight initialization techniques. In terms of convergence speed, we observed that Xavier initialization helps the network converge faster.

We focused on finding the learning capability of the network. To find this, we experimented with the selection of the latent vector. We started with the normal distribution, which ranges from zero to one, and this distribution aids the overall GAN training. We used the spherical uniform distribution  $[-1,1]$ , whose performance is the same as that of the normal distribution. We also found that the Bernoulli distribution (0,1) aids in stable training. One interesting case for the

**TABLE 3.** Summary of the noise vectors results for proposed GAN.

Noise vector choice	Comments
Uniform	Helps the network implement good training
Normal	As good as uniform distribution
Binomial	As good as uniform distribution
Binomial + Sigmoid	Struggles to generate necessary latent values to continue the training
TanH + Normal	Similar to binomial+sigmoid

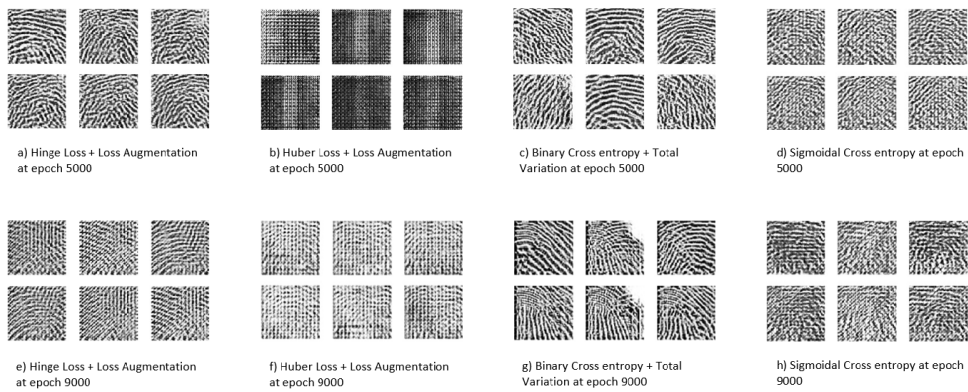
normal distribution is that when it occurs with a nonzero formation, it is superior to the typical normal distribution.

We also applied the activation function ReLU [43], [44], and TanH to the latent vectors. Compared to other choices, ReLU with a spherical uniform distribution shows mixed performance. The hyperbolic tangent version of the Bernoulli distribution showed identical performance as the spherical uniform distribution. Although these experiments show good results, the overall performance depends on the network and the objective function. Our choice of network parameter in the end governs the final outcome. For that reason, we have presented tables 3 and 4 summarizing the compact picture of our total scheme. Figure 2 and 3 is showing the necessary diagrams for the table 2 and 4. The findings in table 2,3, and 4 is obtained through experiments with the proposed network and our input dataset.

Among all the evaluated architectures, we observed the best fingerprint generation training with this architecture. Additionally, we produced full-sized fingerprint images and patch images from this structure. Our discriminator has two versions: with and without spectral normalization. For each model choice, we observed the meaningful generation of synthetic images. The results of this study are presented in the next section.

#### IV. RESULT

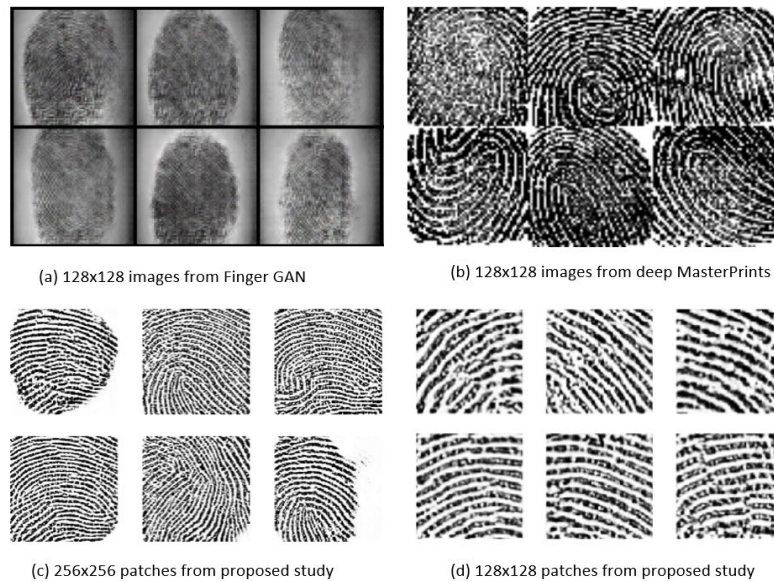
Our study utilizes two different architectures for fingerprint generation. In our network, spectral normalization was used



**FIGURE 3.** The performance of the loss functions, as mentioned in table 2, is present here. Our whole training period is consists of 9000 epochs. Binary Cross entropy + Total Variation loss function seems superior compared to other loss functions but shows inconsistency in training. From figure (b, d, f, h), Sigmoidal cross-entropy, and Huber Loss + loss augmentation was unable to produce meaningful structure over the whole training period. Hinge Loss + loss augmentation is somewhat successful in the first half of the training, and later it degrades the performance of the GAN.

**TABLE 4.** Summary of the proposed network's properties.

Inclusion	Inclusion Detail	Comments
Activations	ReLU, LeakyReLU, TanH, Sigmoid	Sigmoid replacement of ReLU. LeakyReLU performs worst regarding the generator and discriminator.
Batch Normalization	True	For both the generator and discriminator, it helps the network perform well.
Spectral normalization	True	Only for the discriminator. We have tried it for both in every layer. Least design choice is appreciated in our network
Kernel size	5	Between 3 and 5, we have seen better results with 3
Layer Normalization	None	We have tried it for every layer and observed desirable results without it.
Number of filters (generator)	64, 128, 256, 512	
Number of filters (generator)	64, 64, 128, 256, 512	For 256 by 256 size, we have increased one layer.
Number of filters (discriminator)	64, 256, 256, 256	
Number of filters (discriminator)	64, 256, 256, 256	For 256 by 256 size, we have not increased the number of layers.



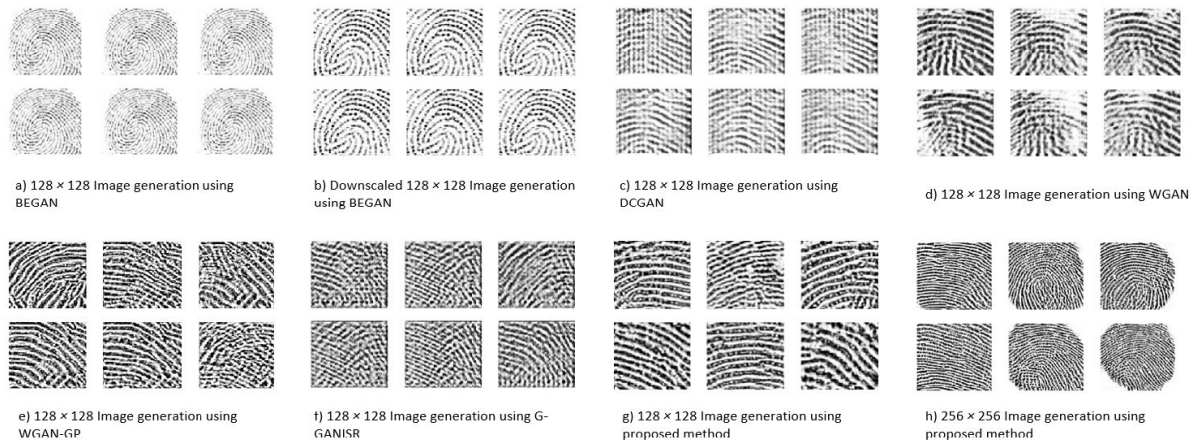
**FIGURE 4.** a) Images from Finger GAN [14]. b) Images from deep MasterPrints [13]. c) 256 × 256 patches from the proposed study. d) 128 × 128 patches from the proposed study. By visual inspection, ridges are clearer and sharper in the images from this study.

for  $128 \times 128$  and  $256 \times 256$  images. We also produced these images without spectral normalization. For this purpose, we used the LivDet fingerprint dataset [27], [45]–[48]. Images in this dataset come with five different scanners. We used images from the Greenbit scanner, which contains

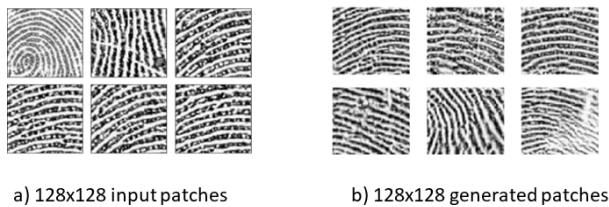
1000 real fingerprints. We applied rotation, translation, and flipping for data augmentation.

Figure 4 shows the output from [13], [14] and the proposed study. Images produced by [14] are blurry compared to those from the other two studies. From figure 4, we can easily

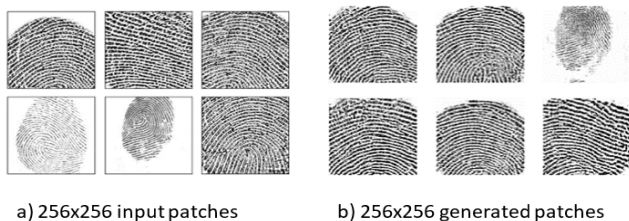




**FIGURE 5.** a)  $128 \times 128$  images with BEGAN [20]. b)  $128 \times 128$  images with BEGAN. Here, input images are  $128 \times 128$  patches downsampled from  $256 \times 256$  patches. For both of these cases, patches are highly similar to each other. BEGAN produces different images with different initializations. However, the amount of diversity is explicitly negligible in all cases for BEGAN. c) Images from the DCGAN where images are somewhat diverse and not fully developed [15]. d)  $128 \times 128$  Images from WGAN where the images are diverse and not fully developed [16]. e)  $128 \times 128$  Images from WGAN-GP where the images are diverse, not fully developed, and comparatively sharper than WGAN [17]. f)  $128 \times 128$  Images from G-GANISR are somewhat recognizable as fingerprints [41]. g)  $128 \times 128$  patches from the proposed study. h)  $256 \times 256$  patches from the proposed study. By visual inspection, ridges are clearer and sharper in the images from the proposed study. Even though BEGAN produced a very stable structure, the amount of diversity is essentially negligible.



**FIGURE 6.** 128 by 128 patch from the input images and the corresponding fake fingerprints.



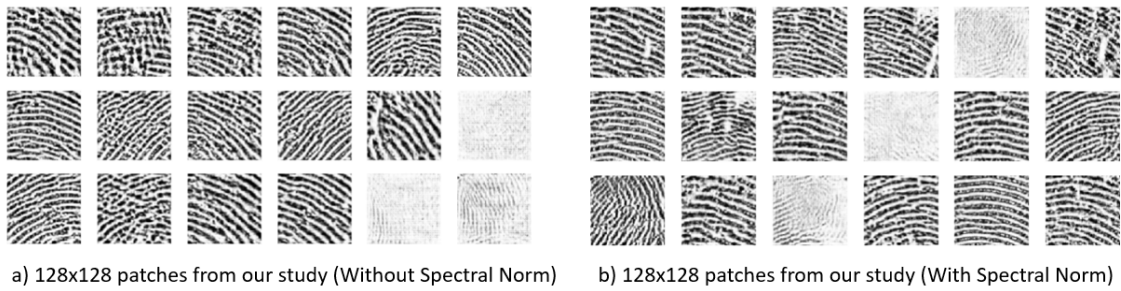
**FIGURE 7.** 256 by 256 patch from the input images and the corresponding fake fingerprints.

observe the ridge difference in the fingerprints. Our study can successfully emulate sharper ridges than the other two studies. Figures 6 and 7 show the input patch images and the respective patches of  $128 \times 128$  and  $256 \times 256$  sizes produced in this study. Figures 8 and 9 contain the images generated using the proposed method. These images show  $128 \times 128$  patches with and without spectral normalization and  $256 \times 256$  patches with and without spectral normalization. Using spectral normalization, we enjoyed a similar quality in the output compared to images without spectral normalization.

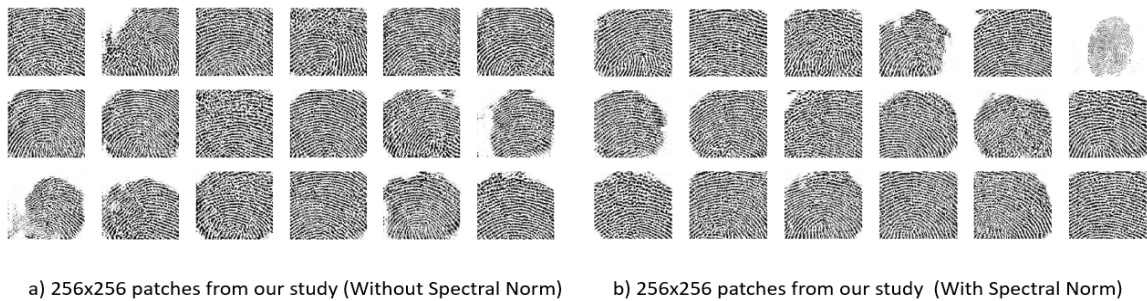
For comparative purposes, we have used the DCGAN [15], WGAN [16], WGAN-GP [17], BEGAN [20], and G-GANISR [41] for image generation. We stacked the output from these in figure 5. BEGAN produces images without any deformity. However, the diversity between generated images is very small. WGAN and DCGAN produce distorted fingerprint images compared to BEGAN. Additionally, these networks provide greater variety in image generation compared to BEGAN. WGAN-GP is somewhat successful in fingerprint generation compared to DCGAN and WGAN. Even though WGAN-GP produce sharper images compared to WGAN, it also struggles to produce desirable fingerprint patches. Images from the G-GANISR performed similar as the DCGAN. All of these methods are viable for producing images with a  $128 \times 128$  size. If we increase the dimension, it is very hard for these methods to produce any meaningful structure. Compared to these methods, our network can produce up to  $256 \times 256$  patches with desirable fidelity and diversity, as shown in figure 5(g-h).

A common way of evaluating the performance of the GAN is to measure the inception distance [17]. Lately, researchers have used the FID score [49] more frequently than the inception distance. The inception score gives us a way to measure the quality of the generated images. This score can be measured using a large number of generated images. The FID score is an improvement on the previously mentioned inception distance. This performance metric compares the statistics of the synthesized images according to the original images. The MS-SSIM [2] score can help us measure the diversity of the generated images. Likewise, this metric also utilizes a large number of generated images. This metric returns a score between 0.0 and 1.0. The higher the score per batch, the lower the amount of diversity among the generated images.

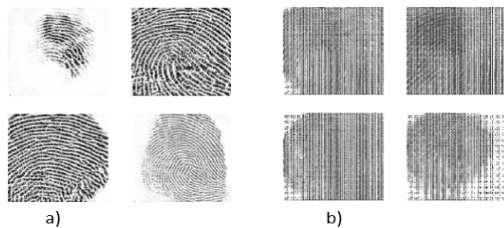




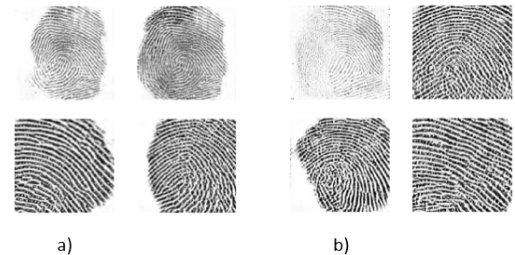
**FIGURE 8.** a) 128 by 128 patch generation. Patches from the proposed network without spectral normalization. b) 128 by 128 patch generation. Patches from the proposed network with spectral normalization.



**FIGURE 9.** a) 256 by 256 patch generation. Patches from the proposed network without spectral normalization. b) 256 by 256 patch generation. Patches from the proposed network with spectral normalization.



**FIGURE 10.** a) Images with loss doping. b) a) Images without loss doping. Left images are from the proposed network at epoch 3244 with the help of loss doping. The images on the right were produced using the same network at epoch 3244 without the aid of loss doping. From these figures, the effect of loss doping on the proposed network is clear.



**FIGURE 11.** a) Images with loss doping. b) Images without loss doping. Images from both sides are at epoch 5975. Here, the proposed network is producing fake fingerprints.

However, we did not use the inception distance or FID score to measure our network performance. These two metrics use weights from the inception network, and these weights are valid for images similar to those in the ImageNet dataset. Since our fingerprint dataset is absent in the ImageNet dataset, it is futile to use these metrics. Hence, we used the MS-SSIM score [2]. This score is entirely different from the other metrics in terms of its application. Since this score does not require an inception network, we can easily use it to measure the performance of our network.

To quantitatively measure our network performance, we used the MS-SSIM metric [2]. Other studies [12]–[14] were not subject to diversity analysis. Table 5 shows the differences between the proposed method and other studies. In this table, [R] stands for the model with a

residual connection, and [S] stands for the model with spectral-residual connections. From table 5, BEGAN shows the lowest diversity performance. Our fingerprint generator achieved better MS-SSIM scores when it came to patches with larger sizes. Since larger size patches contain shapes, ridges, and orientation, it is easier to introduce more diversity.

The MS-SSIM score is empirically lower for patches from the spectral-residual discriminator. This observation is consistent with both small- and large-scale patches. However, the DCGAN and WGAN achieve somewhat good scores even though they seem to produce irregular patches. For BEGAN, the MS-SSIM score is the highest. This result justifies the figure 5. We inserted 128 by 128 patches and downsampled 256 by 256 patches to 128 by 128 patches in the BEGAN network. BEGAN seems to produce the same image every

**TABLE 5. MS-SSIM score comparison.**

Methods	MS-SSIM [2]
BEGAN (patch images, 128 by 128) [20]	0.944
DCGAN (patch images, 128 by 128) [15]	0.5-0.613
WGAN (patch images, 128 by 128) [16]	0.43-0.476
WGAN-GP (patch images, 128 by 128) [17]	0.393-0.441
G-GANISR (patch images, 128 by 128) [41]	0.47-0.558
Proposed method (patch images[S], 128 by 128)	0.31 -0.374
Proposed method (patch images[R], 128 by 128)	0.351-0.425
Proposed method (patch images[S], 256 by 256)	0.23-0.34
Proposed method (patch images[R], 256 by 256)	0.258 -0.32

**TABLE 6. MS-SSIM and average SSIM score comparison.**

Proposed method (MS-SSIM, 128 by 128)	0.31 -0.374
Proposed method (average SSIM, 128 by 128)	0.257-0.324

**TABLE 7. Number of weights.**

Methods	(Generator, Discriminator)
BEGAN [20]	(19.7 M , 3.9 M)
DCGAN [15]	(45.3 M , 7.5 M)
WGAN [16]	(50.3 M , 8.4 M)
WGAN-GP [17]	(50.3 M , 8.4 M)
G-GANISR [41]	(232.4 M , 38.7 M)
Proposed method	(17.8 M , 3.3 M)

time with very slight ridge variation for both of these cases. Moreover, blurred structures are prominent in figure 5 for BEGAN. WGAN-GP produces deformed patches and shows better diversity performance than other methods. G-GANISR shows better diversity than BEGAN and DCGAN.

Compared to all of them, our network achieves greater diversity performance for both 128 by 128 and 256 by 256 patches. Our study has achieved at best 0.23 MS-SSIM for  $256 \times 256$  images with spectral normalization. Without spectral normalization, we achieved a slightly lower score of 0.258 for  $256 \times 256$  images. We also measured SSIM scores between 1000 counterfeit and real images. Thus for one generated image there are 1000 SSIM scores from the real images and we averaged them. We have performed the same for the rest of the fake images. Table 6 contains the average SSIM score information for the 1000 fake fingerprints. This table justifies the MS-SSIM score, since the mean SSIM score of all those GAN generated images did not exceed our achieved MS-SSIM score.

The presented GAN scheme is lighter in terms of trainable weights. For comparison, we have counted number of trainable weights for our network and other studies [15]–[17], [20], [41]. From table 7, we can see that the weight count for our architecture is significantly lower than other state-of-the-art studies.

Usually, the training time required for the GAN is higher than other deep learning networks. Training time for our model varies from 30 hours to several days, depending upon the size of the generated images and the dataset. Our

trained model can produce a batch of 36 fake fingerprints in between 4 to 7 seconds.

## V. CONCLUSION

In this study, we have presented a new GAN scheme to generate fingerprints. The proposed method can successfully produce whole and cropped fingerprint patches with 128 by 128 and 256 by 256 sizes. We have experimentally shown that our network can converge faster with the help of proposed loss doping. Additionally, to generate these fingerprints, our scheme utilizes comparatively a fewer number of weights. Furthermore, our network has demonstrated better divergence performance compared to other state-of-the-art studies. We have also presented experimental results to justify our selection of activation function, noise vector, and network design.

We can easily extend the proposed study for different lines of fingerprint scanners. The proposed doping allows our models to converge faster, although this paper does not cover the generalization of loss doping. We hope this work can provide general insight into the designing of the GAN networks. However, like the current GAN studies, our GAN scheme is not free from a redundant distribution and does not guarantee the deformity free image generation.

In our future work, we would like to extend our research for stable 512 by 512 fingerprint patch generation. Additionally, we are hoping to design a stable GAN architecture that can produce fingerprints with a precise boundary line and high fidelity.

## REFERENCES

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [2] Y. Guo, Q. Chen, J. Chen, Q. Wu, Q. Shi, and M. Tan, "Auto-embedding generative adversarial networks for high resolution image synthesis," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2726–2737, Nov. 2019.
- [3] H. Y. Jung and Y. S. Heo, "Fingerprint liveness map construction using convolutional neural network," *Electron. Lett.*, vol. 54, no. 9, pp. 564–566, May 2018.
- [4] G. L. Marcialis, A. Lewicke, B. Tan, P. Coli, D. Grimbberg, A. Congiu, A. Tidu, F. Roli, and S. Schuckers, "First international fingerprint liveness detection competition—LivDet 2009," in *Proc. Int. Conf. Image Anal. Process.* Berlin, Germany: Springer, 2009, pp. 12–23.
- [5] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli, and S. Schuckers, "Livdet 2011—Fingerprint liveness detection competition 2011," in *Proc. 5th IAPR Int. Conf. Biometrics (ICB)*, 2012, pp. 208–215.
- [6] H. Y. Jung, Y. S. Heo, and S. Lee, "Fingerprint liveness detection by a template-probe convolutional neural network," *IEEE Access*, vol. 7, pp. 118986–118993, 2019.
- [7] A. Roy, N. Memon, J. Togelius, and A. Ross, "Evolutionary methods for generating synthetic MasterPrint templates: Dictionary attack in fingerprint recognition," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 39–46.
- [8] A. Roy, N. Memon, and A. Ross, "MasterPrint: Exploring the vulnerability of partial fingerprint-based authentication systems," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 9, pp. 2013–2025, Sep. 2017.
- [9] R. Cappelli, D. Maio, and D. Maltoni, "Synthetic fingerprint-database generation," in *Proc. Object Recognit. Supported Interact. Service Robot.*, 2002, pp. 744–747.
- [10] P. Johnson, F. Hua, and S. Schuckers, "Texture modeling for synthetic fingerprint generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 154–159.

- [11] Q. Zhao, A. K. Jain, N. G. Paulter, and M. Taylor, "Fingerprint image synthesis based on statistical feature models," in *Proc. IEEE 5th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2012, pp. 23–30.
- [12] K. Cao and A. Jain, "Fingerprint synthesis: Evaluating fingerprint search at scale," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 31–38.
- [13] P. Bontrager, A. Roy, J. Togelius, N. Memon, and A. Ross, "DeepMasterPrints: 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.
- [14] S. Minaee and A. Abdolrashidi, "Finger-GAN: Generating realistic fingerprint images using connectivity imposed GAN," 2018, *arXiv:1812.10482*. [Online]. Available: <http://arxiv.org/abs/1812.10482>
- [15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: <http://arxiv.org/abs/1511.06434>
- [16] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: <http://arxiv.org/abs/1701.07875>
- [17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
- [18] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of GANs," 2017, *arXiv:1705.07215*. [Online]. Available: <http://arxiv.org/abs/1705.07215>
- [19] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?" 2018, *arXiv:1801.04406*. [Online]. Available: <http://arxiv.org/abs/1801.04406>
- [20] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*. [Online]. Available: <http://arxiv.org/abs/1703.10717>
- [21] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*. [Online]. Available: <http://arxiv.org/abs/1802.05957>
- [22] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier Gans," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2642–2651.
- [23] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*. [Online]. Available: <http://arxiv.org/abs/1809.11096>
- [24] T. Nguyen, T. Le, H. Vu, and D. Phung, "Dual discriminator generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2670–2680.
- [25] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," 2016, *arXiv:1611.01673*. [Online]. Available: <http://arxiv.org/abs/1611.01673>
- [26] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti, "Stabilizing GAN training with multiple random projections," 2017, *arXiv:1705.07831*. [Online]. Available: <http://arxiv.org/abs/1705.07831>
- [27] B. Zhao, W. Li, and W. Gong, "Deep pyramid generative adversarial network with local and nonlocal similarity features for natural motion image deblurring," *IEEE Access*, vol. 7, pp. 185893–185907, 2019.
- [28] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "Adagan: Boosting generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5424–5433.
- [29] A. Ghosh, V. Kulharia, V. Nambodiri, P. H. S. Torr, and P. K. Dokania, "Multi-agent diverse generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8513–8521.
- [30] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung, "MGAN: Training generative adversarial nets with multiple generators," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada. Amherst, MA, USA: OpenReview, 2018.
- [31] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5907–5915.
- [32] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.
- [33] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*. [Online]. Available: <http://arxiv.org/abs/1805.08318>
- [34] W. Lee, D. Kim, S. Hong, and H. Lee, "High-fidelity synthesis with disentangled representation," 2020, *arXiv:2001.04296*. [Online]. Available: <http://arxiv.org/abs/2001.04296>
- [35] A. Karnewar, O. Wang, and R. S. Iyengar, "MSG-Gan: Multi-scale gradient gan for stable image synthesis," *CoRR*, vol. abs/1903.06048, Jun. 2019.
- [36] Y. Miron and Y. Coscas, "S-flow GAN," 2019, *arXiv:1905.08474*. [Online]. Available: <http://arxiv.org/abs/1905.08474>
- [37] C. H. Lin, C.-C. Chang, Y.-S. Chen, D.-C. Juan, W. Wei, and H.-T. Chen, "COCO-GAN: Generation by parts via conditional coordinating," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4512–4521.
- [38] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*. [Online]. Available: <http://arxiv.org/abs/1710.10196>
- [39] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2019, pp. 4401–4410.
- [40] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.
- [41] P. Shamsolmoali, M. Zareapoor, R. Wang, D. K. Jain, and J. Yang, "G-GANISR: Gradual generative adversarial network for image super resolution," *Neurocomputing*, vol. 366, pp. 140–153, Nov. 2019.
- [42] M. Zareapoor, H. Zhou, and J. Yang, "Perceptual image quality using dual generative adversarial network," *Neural Comput. Appl.*, pp. 1–11, May 2019.
- [43] W. Zhou, C. Wu, Y. Yi, and G. Luo, "Structure preserving non-negative feature self-representation for unsupervised feature selection," *IEEE Access*, vol. 5, pp. 8792–8803, 2017.
- [44] Z. Qiumei, T. Dan, and W. Fenghua, "Improved convolutional neural network based on fast exponentially linear unit activation function," *IEEE Access*, vol. 7, pp. 151359–151367, 2019.
- [45] *LivDet Liveness Detection Competition Series*. Accessed: Feb. 2020. [Online]. Available: <http://livdet.org>
- [46] L. Ghiani, D. A. Yambay, V. Mura, G. L. Marcialis, F. Roli, and S. A. Schuckers, "Review of the fingerprint liveness detection (LivDet) competition series: 2009 to 2015," *Image Vis. Comput.*, vol. 58, pp. 110–128, Feb. 2017.
- [47] V. Mura, G. Orru, R. Casula, A. Sibiriu, G. Loi, P. Tuveri, L. Ghiani, and G. L. Marcialis, "LivDet 2017 fingerprint liveness detection competition 2017," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 297–302.
- [48] L. Ghiani, D. Yambay, V. Mura, S. Tocco, G. L. Marcialis, F. Roli, and S. Schuckers, "LivDet 2013 fingerprint liveness detection competition 2013," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–6.
- [49] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.



**MASUD AN-NUR ISLAM FAHIM** received the B.Sc. degree in electronics and communication engineering from the Khulna University of Engineering and Technology (KUET), Khulna, Bangladesh, in 2017. He is currently pursuing the master's degree with Chosun University. From 2017 to 2018, he was affiliated with electronic industry, and later he joined the academy. He is interested in deep learning theory, computer vision, machine learning, optimization theory, and computational photography.



**HO YUB JUNG** received the B.S. degree in electrical engineering from The University of Texas at Austin, in 2002, and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, in 2006 and 2012, respectively. He was with Samsung Electronics for two years, as a Senior Engineer. Since 2017, he has been an Assistant Professor with the Department of Computer Engineering, Chosun University. His research interests include computer vision, machine learning, and medical imaging.

...