Gradient-Penalty GAN Framework for High-Fidelity Fingerprint Synthesis

Oleksandr Striuk [1,2], Yuriy Kondratenko [1,3]

[1] Petro Mohyla Black Sea National University, 68 Desantnykyv St. 10, Mykolaiv, 54000, Ukraine
[2] University of Ostrava, Department of Mathematics, Dvořákova 7, Ostrava, 70103, Czech Republic
[3] Institute of AI Problems under MES and NAS of Ukraine, str. Mala Zhytomyrska, 11, office 5a, Kyiv, 01001, Ukraine

Abstract

The rapid advancements in generative adversarial networks (GANs) have significantly impacted digital content synthesis, presenting both opportunities and challenges in multimedia forensics and cybersecurity. We present an Enhanced Adaptive DCGAN (EADC-GAN) for generating high-fidelity synthetic fingerprints, addressing core challenges in training stability and sample diversity. By combining Wasserstein loss with gradient penalty (WGAN-GP), instance normalization in the discriminator, and tailored architectural refinements, our model achieves strong image realism at reduced training cost. Compared to prior DCGAN-based methods, EADC-GAN synthesizes more diverse, artifact-free samples in fewer epochs, making it suitable for scalable biometric data generation. This has key implications for secure authentication, privacy-preserving biometric datasets, and adversarial robustness in cybersecurity contexts.

Keywords

Synthetic fingerprints, generative adversarial networks, WGAN-GP, DCGAN, instance normalization, biometric security, adversarial robustness, deep learning, cybersecurity, fingerprint synthesis

1. Introduction

In the modern era, artificial intelligence (AI) and information technology (IT) have become deeply embedded in nearly every aspect of human activity, driving advancements in automation, decision-making, and security. From smart homes to autonomous systems, AI-powered solutions enhance efficiency and enable novel applications across industries. In particular, AI has revolutionized forensic investigations and secure access control systems, where accurate and reliable identification methods are crucial [7, 30, 32].

Fingerprint-based biometric systems have become a cornerstone of modern security infrastructures, owing to their robustness and uniqueness in identifying individuals. With applications ranging from smartphone authentication to large-scale national identity programs, the demand for high-quality, reliable fingerprint data has soared.

However, the collection of large-scale, diverse, and privacy-preserving fingerprint datasets can be both resource-intensive and ethically fraught. This challenge has prompted research into synthetic fingerprint generation methods that can provide abundant, high-fidelity data without exposing sensitive personal information [1, 2].

Generative Adversarial Networks (GANs) have emerged as efficient tools for synthetic and realistic data generation, offering compelling results in various domains including image, video, and audio synthesis. Despite their success, early GAN models often suffered from training instabilities and mode collapse, limiting their applicability to more sensitive tasks such as fingerprint synthesis [3]. Variations like Deep Convolutional GANs (DCGANs) introduced architectures tailored for image generation, yet challenges remained, particularly when targeting both high quality and diversity in the generated outputs [4].

One promising improvement to the GAN framework is the use of gradient penalty techniques, such as those found in Wasserstein GANs with Gradient Penalty (WGAN-GP), which offer enhanced training stability. Additionally, normalization layers have a profound impact on the training dynamics and generation quality of GANs. Adaptive instance normalization (AIN), for example, has shown the capability to improve style consistency and reduce artifacts in image synthesis tasks [5, 6].

[1]CMIS-2025: Eighth International Workshop on Computer Modeling and Intelligent Systems, May 5, 2025, Zaporizhzhia, Ukraine

oleksandr.striuk@gmail.com (O. Striuk); yuriy.kondratenko@chmnu.edu.ua (Y. Kondratenko)
0000-0002-6391-4382 (O. Striuk); 0000-0001-7736-883X (Y. Kondratenko)

Early approaches to fingerprint image synthesis explored traditional models such as DCGAN. In our previous work, Adaptive Deep Convolutional GAN for Fingerprint Sample Synthesis (ADCGAN), we demonstrated that a well-tuned DCGAN could generate visually convincing fingerprints [7]. However, the method exhibited two main drawbacks. First, achieving photorealistic fingerprints demanded a large number of epochs — often exceeding 1,000 — to reach acceptable quality. Second, despite eventually producing samples with realistic ridge patterns, the model became prone to mode collapse at higher epoch counts. This collapse led to repetitive samples and diminished the overall diversity of the generated dataset. The need for extensive training time also poses challenges for projects with limited computational resources or time-sensitive development cycles [7].

In this paper, we present a Gradient-Penalty GAN Framework for high-fidelity fingerprint synthesis, leveraging insights from both WGAN-GP and enhanced normalization strategies [8, 9, 10]. We build upon the insights gained from ADCGAN, refining the architecture and training strategy to yield higher-quality samples in fewer epochs while minimizing mode collapse.

By optimizing the instance normalization layers and incorporating gradient penalty, we aim to address training instability and mode diversity issues that often hinder fingerprint GAN models. Our experimental evaluations demonstrate that this architecture not only produces more realistic and varied fingerprint images but also reduces the computational overhead commonly associated with GAN enhancements.

Crucially, the synthesized fingerprints can bolster biometric research by providing large-scale datasets and facilitating the development of advanced, secure authentication systems without compromising user privacy.

Despite advances in GAN-based fingerprint synthesis, existing models often struggle with training instability, mode collapse, and insufficient diversity in generated samples. Additionally, many methods require extensive computational resources and training time, limiting their practical use in scalable biometric systems. This work addresses these limitations by proposing a more stable, efficient GAN framework capable of producing realistic, high-resolution fingerprint images with minimal redundancy.

2. Related Work

2.1. GAN-Based Image Synthesis Approaches

GANs have become a central method for synthesizing diverse image datasets, including biometric images such as fingerprints. Early efforts often relied on the DCGAN framework, which demonstrated that transposed convolution layers could capture essential fingerprint patterns. However, these classical DCGANs typically require extensive training, and they remain susceptible to mode collapse — where the generator converges to limited variations of the same fingerprint. Recent approaches have aimed to mitigate these issues by integrating more advanced loss functions and architectural refinements, making GAN-based finger-print synthesis both more efficient and more robust in capturing fine ridge details [5, 6].

2.2. Fingerprint Synthesis Techniques

Early fingerprint generation efforts often employed parametric and procedural models, focusing on ridge flow simulation and minutiae placement through mathematical functions. Techniques such as Gabor-based filters, Fourier transforms, and partial differential equations (PDEs) aimed to replicate key fingerprint structures without relying on large training sets. Although these methods can yield convincing ridge patterns and minutiae distributions, they sometimes lack the capacity to produce the extensive variability needed for modern biometric applications.

In contrast, deep learning–driven approaches like GANs learn distributional properties directly from real data, offering greater flexibility and diversity in synthesized outputs. Beyond standard DCGAN-based solutions, advanced architectures — such as StyleGAN and CycleGAN — further refine texture details, enhance global coherence, and address common pitfalls like mode collapse. Together, both classical (model-based) and deep learning–based techniques enrich the toolbox for generating comprehensive, privacy-friendly fingerprint datasets [7, 11, 12].

2.3. Normalization in GANs

Normalization layers are crucial for stabilizing GAN training, and adaptive instance normalization offers particular benefits for image synthesis tasks that depend on local texture fidelity. While batch normalization averages statistics across a mini-batch, instance normalization normalizes each sample independently, helping preserve distinctive ridges and fine details in synthetic fingerprints. Instance normalization can reduce style variations within a single batch — an advantage when the primary goal is to maintain consistent textural cues. As a result, integrating instance normalization, especially in the discriminator, can sharpen feature detection and further mitigate common GAN pitfalls such as training instability and overly uniform outputs [13, 14].

2.4. Gradient Penalty Methods (WGAN-GP and Beyond)

The Wasserstein GAN (WGAN) framework addresses two key shortcomings in conventional GANs: vanishing gradients and unstable training. By replacing the standard generator-discriminator loss with the Wasserstein distance, WGAN provides a meaningful gradient signal that promotes better convergence. To further stabilize training, WGAN with Gradient Penalty (WGAN-GP) incorporates a gradient penalty term that enforces Lipschitz continuity without resorting to weight clipping. This penalty term substantially reduces mode collapse and improves sample diversity. In the context of fingerprint synthesis, WGAN-GP's training stability helps produce more varied and realistic fingerprint ridges over fewer epochs [8, 9].

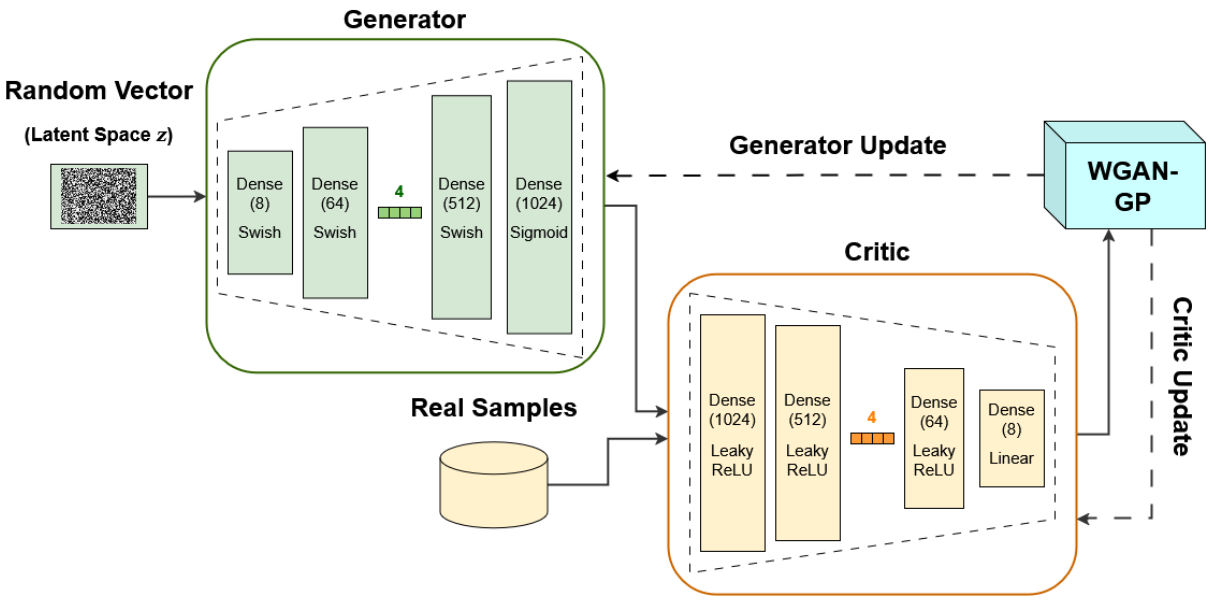

Figure 1: Flowchart illustrating the workflow of the WGAN-GP architecture

Unlike classical DCGAN-based approaches that rely heavily on batch normalization and often exhibit mode collapse, our method introduces instance normalization in the discriminator and leverages WGAN-GP for improved gradient flow. This combination enhances both training stability and output diversity. While models like StyleGAN and CycleGAN achieve high visual fidelity, they often require complex tuning and are not specifically tailored to biometric features [5, 15, 17]. In contrast, our architecture is optimized for fingerprint synthesis, balancing computational efficiency with domain-specific texture preservation.

3. Proposed Methodology

3.1. Normalized DCGAN Architecture

The proposed model builds upon the standard DCGAN framework — originally designed to generate high-quality images through transposed convolutions in the generator and strided convolutions in the discriminator.

However, instead of using batch normalization throughout, we use instance normalization in the discriminator to improve training stability and capture finer textures critical for biometric features.

DCGAN was chosen as the foundation due to its proven effectiveness in structured image generation, including biometric textures. Its simplicity and modularity make it highly adaptable for fingerprint synthesis. By replacing batch normalization with instance normalization and integrating WGAN-GP, we retain DCGAN's strengths while resolving its typical weaknesses — namely, training instability and low sample diversity. This adapted framework strikes a practical balance between architectural simplicity, computational efficiency, and output quality.

Generator follows a classic DCGAN-like upsampling pipeline, which transforms a latent noise vector into a full-resolution fingerprint image through stacked transposed convolutional layers, batch normalization, and ReLU activations.

Discriminator mirrors the generator's structure in a downsampling fashion but replaces batch normalization with instance normalization layers. LeakyReLU activations are retained to preserve gradient flow. Instead of normalizing over the entire batch, instance normalization (IN) scales and shifts each sample independently. Fingerprint images demand precise ridge patterns. IN helps retain such fine-grained textures without inadvertently averaging them out across a mini-batch [13, 15, 16, 17].

By combining instance normalization with the WGAN-GP training strategy, the model is less prone to collapsing to repetitive samples. Empirical observations show that instance normalization can better capture local variations, which are paramount for realistic fingerprint synthesis. In the following sections, we detail how gradient penalty is integrated to further stabilize training and discuss the specific loss functions and optimization protocol that tie into the model architecture.

Table 1
Model Configuration

Layer Type	Configurations
Fully Connected (Generator)	#units: 512×8×8
Reshape & BatchNorm	#features: 512, BN, ReLU
Transposed Convolution	#filters: 256, k: 4×4, s: 2, p: 1
BatchNormalization + ReLU	#features: 256, BN, ReLU
Transposed Convolution	#filters: 128, k: 4×4, s: 2, p: 1
BatchNormalization + ReLU	#features: 128, BN, ReLU
Transposed Convolution	#filters: 64, k: 4×4, s: 2, p: 1
BatchNormalization + ReLU	#features: 64, BN, ReLU
Transposed Convolution (Output)	#filters: 1, k: 4×4, s: 2, p: 1, activation: Tanh
Input (Discriminator)	1×128×128 grayscale images
Convolution	#filters: 64, k: 4×4, s: 2, p: 1, activation: LeakyReLU(0.2)
Convolution + InstanceNorm	#filters: 128, k: 4×4, s: 2, p: 1, activation: LeakyReLU(0.2)
Convolution + InstanceNorm	#filters: 256, k: 4×4, s: 2, p: 1, activation: LeakyReLU(0.2)
Convolution + InstanceNorm	#filters: 512, k: 4×4, s: 2, p: 1, activation: LeakyReLU(0.2)
Convolution (Output)	#filters: 1, k: 4×4, s: 1, p: 0, activation: Linear

3.2. Integration of Gradient Penalty

To stabilize training and mitigate mode collapse, we adopt the WGAN-GP framework, which enforces Lipschitz continuity through a gradient penalty on the discriminator's output. Unlike weight clipping used in early WGANs, this penalty regularizes gradient norms for interpolated real and fake samples, improving convergence without harming model capacity [8, 9].

In each training iteration, the algorithm samples a random scalar α from a uniform distribution $U(0,1)$. A point $\hat{x}$ is then created by interpolating between a real sample x_{real} and a generated sample x_{fake}. The discriminator's gradient is computed on $\hat{x}$.

Let's formulate the loss. If D denotes the discriminator, its Wasserstein distance–based objective incorporates an added penalty term:

$$\lambda_{gp} \cdot (\| \nabla D(\hat{x}) \|_2 - 1)^2 \tag{1}$$

where λ_{gp} is a hyperparameter dictating the penalty's strength.

By penalizing large deviations of $\| \nabla D(\hat{x}) \|_2$ from 1, the discriminator remains closer to a valid 1-Lipschitz function, leading to more reliable gradients for the generator [8, 9].

The gradient penalty term mitigates abrupt updates in the discriminator that commonly cause training to diverge. By preserving a stable gradient flow, the generator avoids collapsing to a narrow subset of fingerprints. Because the discriminator's updates remain well-conditioned, the model can converge to realistic fingerprint patterns in fewer epochs compared to weight-clipped or standard DCGAN setups [8, 9].

In the next sections, we detail how this WGAN-GP loss formulation is combined with instance normalization, specialized generator and discriminator architectures, and the overall training workflow to produce high-quality, diverse fingerprint images.

3.3. Loss Functions and Optimization Strategy

The training process adopts the WGAN-GP framework, which replaces the traditional adversarial loss with an objective based on the Wasserstein distance. Below are the key components [8, 9].

Discriminator (Critic) Loss:

$$L_D = E_{x \sim p_{data}}[D(x)] - E_{z \sim p_z}[D(G(z))] + \lambda_{gp} E_{\hat{x}}[(\| \nabla_{\hat{x}} D(\hat{x}) \|_2 - 1)^2] \tag{2}$$

where L_D is the discriminator loss, E is expectation, D is the discriminator, G is the generator, x are real samples, z are noise vectors from a prior distribution (e.g., $N(0,1)$), $\hat{x}$ is an interpolated sample between real and generated data, and λ_{gp} scales the gradient penalty.

Generator Loss:

$$L_G = -E_{z \sim p_z}[D(G(z))] \tag{3}$$

where L_G is the generator loss.

Adam is employed for both generator and discriminator, with learning rate $\approx 2 \times 10^{-4}$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. These parameters promote stable convergence in convolutional architectures, particularly with the gradient penalty term.

A common strategy in WGAN-based setups is to update the discriminator more often than the generator (e.g., 5:1 ratio), ensuring the critic remains sufficiently accurate to guide generator updates.

Gradient penalty enforces a smooth, 1-Lipschitz constraint without resorting to weight clipping. Instance normalization in the discriminator helps preserve detailed ridge features in fingerprints and ensures stable gradient flow [8, 9].

By combining WGAN-GP loss functions, careful hyperparameter tuning, and selective update frequencies, the model converges faster and produces higher-fidelity fingerprint images than standard DCGAN-based methods.

3.4. Algorithmic Workflow

The training pipeline, informed by the enhanced EADC-GAN implementation, proceeds through a structured sequence of steps designed to systematically refine both the generator and discriminator networks. Initially, the model configurations and hyperparameters are defined, including the number of epochs, batch size, latent dimension, learning rate, and the gradient penalty coefficient. Fingerprint images, resized to 128×128 pixels and normalized to the [−1,1], are loaded through a shuffling mechanism that ensures an unbiased sampling process across mini-batches.

Once the data is loaded, the generator and discriminator networks are initialized. The generator uses DCGAN-like architecture to transform a latent vector $z \in R^{100}$ into a 128×128 image. It applies transposed convolutions, batch normalization, ReLU activations, and ends with a Tanh layer. The discriminator mirrors a downsampling approach, incorporating instance normalization and LeakyReLU activations. To enable stable convergence, both networks initialize their learnable parameters with random values drawn from a normal distribution centered at zero with a standard deviation of 0.02.

The core training cycle repeats for each epoch and processes one mini-batch of fingerprint data at a time. In each iteration, the discriminator is first updated by sampling real fingerprint images x_{real} from the dataset and generating fake images $x_{fake} = G(z)$ from randomly sampled noise vectors z. The Wasserstein distance is then computed as the difference in discriminator outputs on real and fake samples, and the gradient penalty term is imposed through an interpolation strategy that regularizes the norm of the discriminator's gradients. These gradient-based objectives are combined, and the discriminator parameters are updated accordingly via backpropagation with an Adam optimizer, using momentum parameters $\beta = (0.5, 0.999)$.

After the discriminator update, the generator is refined at a reduced frequency (for instance, every five discriminator iterations) to maintain a reliable critic. In this phase, fresh noise vectors are drawn from the latent distribution, passed through the generator, and evaluated by the discriminator. The generator's loss function aims to maximize the discriminator's output on these synthesized images, effectively minimizing the negative Wasserstein distance. By backpropagating this signal, the generator weights are adjusted to create more plausible and diverse fingerprint images in subsequent iterations.

Throughout training, the system periodically saves both model checkpoints and synthetic images generated from a fixed set of noise vectors. These outputs allow for consistent evaluation of the generator's progression over time and facilitate direct comparison across epochs. Upon completion, the final weights of the generator and discriminator are stored for downstream usage, such as bulk synthetic fingerprint generation or further fine-tuning. By blending frequent discriminator updates, a gradient penalty mechanism, and controlled generator refinement, the workflow produces high-fidelity and structurally diverse fingerprint images within a stable and computationally efficient training regime.

Below is a comparison table that presents architectural differences between the models — our previous ADC-GAN and EADC-GAN [7].

3.1. Theoretical Contribution and Novelty

This work presents a novel synthesis of two stabilizing strategies — gradient penalty from WGAN-GP and instance normalization in the discriminator — to improve the fidelity and diversity of fingerprint generation. While both techniques have been explored separately in GAN literature, their combined use and fine-tuning in a domain-specific architecture for fingerprint synthesis is new, and, to our knowledge, no prior fingerprint-synthesis study combines IN and WGAN-GP. Our results demonstrate that this hybrid strategy enables faster convergence, reduces mode collapse, and preserves fine-grained biometric details more effectively than conventional batch-normalized DCGANs or standard WGAN-GP models.

Table 2
Comparison Table of ADC-GAN and EADC-GAN

Feature	ADC-GAN	EADC-GAN
Image Size	64 × 64 pixels (images are resized and center-cropped to 64×64).	128 × 128 pixels (images are resized to 128×128).
Latent Dimension	100 (input noise vector size is 100).	100 (input noise vector size is 100).
Training Epochs	1200 epochs.	200 epochs.
Loss Function	GAN loss and Binary Cross-Entropy (BCE) loss for both generator and discriminator.	Wasserstein GAN loss with gradient penalty (WGAN-GP): the discriminator loss is computed as the negative difference between real and fake outputs plus a gradient penalty term ($\lambda = 10$).
Normalization (G)	Batch normalization is used after each deconvolution (ConvTranspose2d) layer except the output layer.	Batch normalization is applied in the generator after the fully connected layer and during deconvolution.
Normalization (D)	Batch normalization is applied (after the first Conv2d layer).	Instance normalization is used instead of batch normalization in most layers, which may help stabilize training when combined with gradient penalty.
Optimizer & Learning Rate	Adam optimizer with separate learning rates: Generator lr = 0.0001, Discriminator lr = 0.0002; $\beta_1 = 0.5$, $\beta_2 = 0.999$.	Adam optimizer for both networks with a common learning rate of 0.0002; $\beta_1 = 0.5$, $\beta_2 = 0.999$.
Generator Architecture	A sequential model that uses a series of ConvTranspose2d layers to upsample the latent vector directly into a 64×64 image; final activation is Tanh.	Starts with a fully connected (fc) layer that projects the latent vector into a feature map (reshaped to an 8×8 spatial size) followed by several ConvTranspose2d layers to upscale to 128×128; final activation is Tanh.
Discriminator Architecture	A sequential network of Conv2d layers with BatchNorm (except the first layer) and LeakyReLU activations; ends with a Sigmoid to output a probability score.	A sequential network of Conv2d layers using instance normalization and LeakyReLU activations; does not use a Sigmoid activation in the final layer – it outputs a single scalar value for WGAN-based loss calculation.
Unique Features	Implements a modified DCGAN architecture with BCE loss, adaptive learning rates, robust weight initialization, and batch normalization techniques.	Incorporates instance normalization in the discriminator, modified batch normalization, and a gradient penalty term in the loss (WGAN-GP), improving training stability, computational effectiveness, and image quality in higher resolution settings.

Furthermore, we extend the practical utility of WGAN-GP by applying it to a biometric domain with strict texture preservation needs, showing its scalability to higher resolutions (128×128) with reduced training costs. This framework can serve as a foundational baseline for synthetic biometric data generation, adversarial robustness studies, and privacy-focused authentication system design.

4. Experimental Setup

4.1. Datasets and Preprocessing

The experimental analysis utilizes a curated subset of fingerprint images drawn from a publicly available biometric dataset — SOCOFing [18]. The data includes grayscale samples, exhibit variability in ridge patterns, and contrast.

To achieve consistency across samples, each fingerprint image is resized to a fixed spatial dimension of 128×128 pixels and normalized to the range [−1,1]. This normalization aligns with the output of the generator's Tanh activation, facilitating stable training dynamics and seamless comparisons across different batches.

The grayscale format (single-channel) not only reduces computational overhead but also highlights finer ridge and valley structures, which are central to realistic fingerprint generation. Throughout the preprocessing pipeline, data is split into training and validation subsets, although the adversarial framework primarily relies on the training partition for iterative updates.

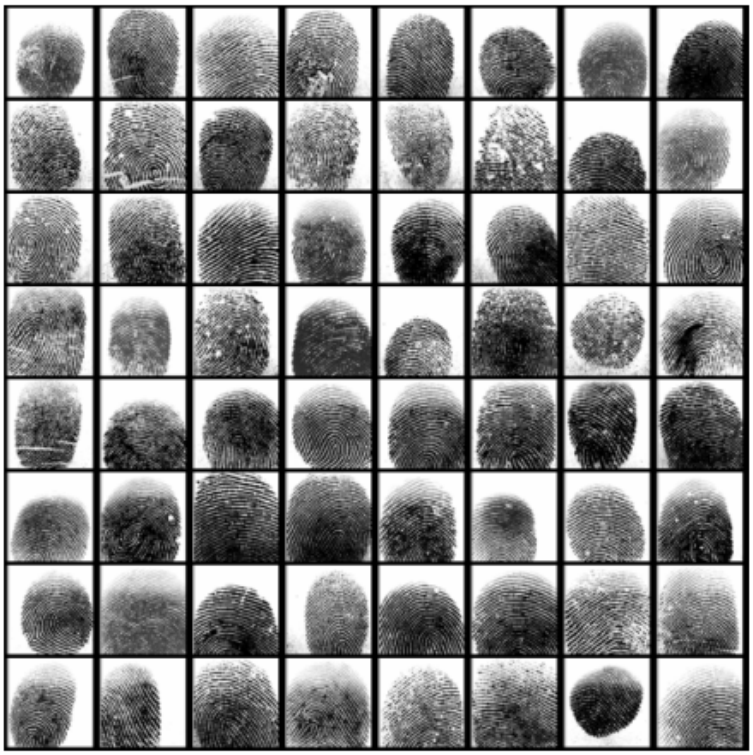

Figure 2: SOCOFing real data — fingerprint scans

4.2. Evaluation Metrics

The model monitors training progress primarily through discriminator and generator loss values, as well as periodic visual inspection of generated samples. By regularly printing the Wasserstein-based objective for both networks, we were able to quickly identify training instabilities. Meanwhile, saving a fixed batch of synthetic fingerprint images over multiple epochs provides a direct, qualitative perspective on improvements in ridge fidelity and overall realism.

Relying on losses and sample outputs offers a lightweight yet effective evaluation strategy. In high-detail domains like fingerprint synthesis, real-time visual checks can be more intuitive than abstract numeric scores, enabling domain experts to spot subtle artifacts or textural inconsistencies. This

approach also simplifies model development by reducing the computational overhead of advanced metrics (e.g., Fréchet Inception Distance), which often require large external classifiers or additional memory usage.

If a more robust, quantitative benchmark is desired, metrics like FID or Inception Score can be incorporated at a later stage to complement the qualitative insights gained from losses and sample images.

4.3. Implementation Details and Hyperparameter Settings

Model training is conducted using PyTorch, leveraging GPU acceleration. Both generator and discriminator networks are initialized with weights drawn from a normal distribution ($\mu=0$, $\sigma=0.02$), consistent with DCGAN best practices. The Adam optimizer [19] is applied to each network's parameters.

The discriminator (equipped with instance normalization) is updated for every mini-batch, while the generator receives updates at a slightly reduced frequency (e.g., once every five discriminator steps), preserving a balanced training signal.

The gradient penalty coefficient λ_{gp} is set to 10, based on prior WGAN-GP literature that suggests it effectively constrains the gradient norm [8, 9].

Training proceeds for up to 200 epochs — substantially fewer than the 1,000+ epochs sometimes required by earlier DCGAN-based methods — owing to the stabilizing influence of gradient penalty and the enhanced texture preservation afforded by instance normalization.

5. Results and Analysis

5.1. Qualitative Assessment

Generated fingerprint samples display high-fidelity ridge patterns and minimal visual artifacts, particularly in mid-to-late training epochs. Qualitatively, synthetic images exhibit distinct papillary lines, consistent contrast levels, and plausible global orientations that resemble real biometric data. By periodically saving and reviewing the generator's outputs, we could observe a steady progression from coarse, noisy impressions to well-defined ridge structures.

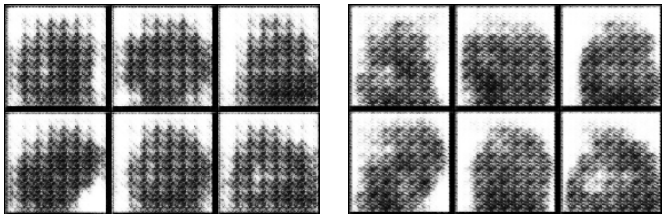

Figure 3: Synthesized samples — results after 10 epochs (left) and 20 epochs (right) of EADC-GAN model training

Notably, improvements occur with only 170—200 epochs compared to 1200 epochs of the earlier DCGAN-based model, reflecting the stabilizing influence of the gradient penalty and instance normalization.

5.2. Normalization and Penalty Variants

Ablation experiments indicate that substituting batch normalization with instance normalization in the discriminator enhances texture preservation and mitigates mode collapse. When batch normalization is reintroduced, training exhibits higher variance in discriminator loss and a slight decline in sample diversity. Similarly, reducing or removing the gradient penalty coefficient (λ_{gp}) increases the likelihood of training instabilities and partially reintroduces repetitive patterns in generated outputs.

These findings confirm that both instance normalization and a carefully tuned gradient penalty are key contributors to generating varied fingerprints.

5.3. Computational Efficiency and Scalability

Despite incorporating gradient penalty and additional normalization layers, the model proves computationally efficient relative to extended training regimes of-ten required by baseline DCGANs.

In practical experiments, fewer total epochs are needed to attain comparable — or superior — visual fidelity. Moreover, the approach scales well on standard GPU hardware, supporting batch sizes large enough to accelerate convergence. This efficiency stems from the stable gradient updates afforded by WGAN-GP, which reduce the need for extensive hyperparameter searches and lessen the risk of early divergence, making the framework suitable for larger datasets or more complex biometric tasks.

5.4. Evaluation of Results

The training dynamics of the model exhibit an initial phase of instability, particularly in the discriminator's loss, which starts at an excessively high value in the first epoch. This behavior suggests that the gradient penalty term may have been dominating due to scaling. However, within the first few epochs, the discriminator loss rapidly decreases and stabilizes around -0.5 to -0.8, indicating that the discriminator quickly adapts to distinguishing real from generated samples. Simultaneously, the generator loss begins at a low value and progressively increases, demonstrating an initial struggle to generate realistic samples. By epochs 10–20, the adversarial balance improves, as evidenced by the increasing generator loss and stabilized discriminator loss, suggesting that the generator is effectively learning to produce more convincing outputs.

Below are samples of artificial fingerprints generated by EADC-GAN after 170 and 200 epochs, respectively.

Figure 4: Synthesized fingerprints — results after 170 epochs (left) and 200 epochs (right) of EADC-GAN model training

For comparison, the results after 1,000 to 1,300 training epochs are presented below. The samples clearly demonstrate mode collapse, along with low-resolution quality.

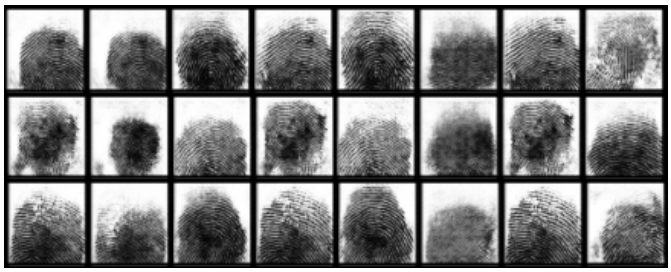

Figure 5: Generated results obtained from 1000 to 1300 training epochs

Beyond epoch 40, the training stabilizes further, with the generator loss continuing to rise and reaching values around 4.0 by epoch 100. This steady increase indicates that the generator is persistently improving its ability to generate high-quality images, while the discriminator maintains a controlled dominance. The gradient penalty (λ=10) appears to regulate training effectively, preventing extreme discriminator outputs and ensuring stable adversarial interactions. However, the increasing generator loss may warrant further investigation to rule out potential training inefficiencies or diminishing discriminator feedback. Overall, the observed loss trends suggest that the

model is learning effectively, but parameter tuning — especially for the gradient penalty coefficient — could further optimize convergence dynamics.

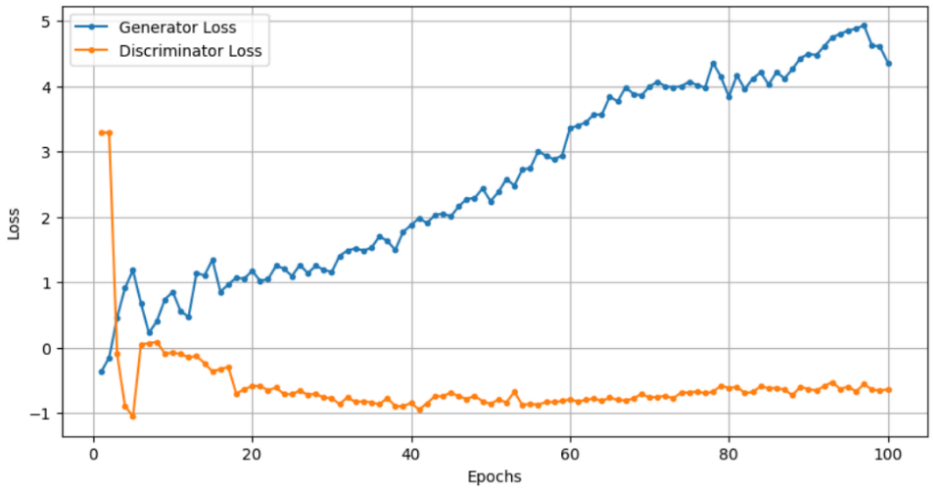

Figure 6: Generator and discriminator loss during training.

The model was trained using the Kaggle cloud environment with an NVIDIA Tesla P100-PCIE-16GB GPU, running CUDA 12.6 and driver version 560.35.03.

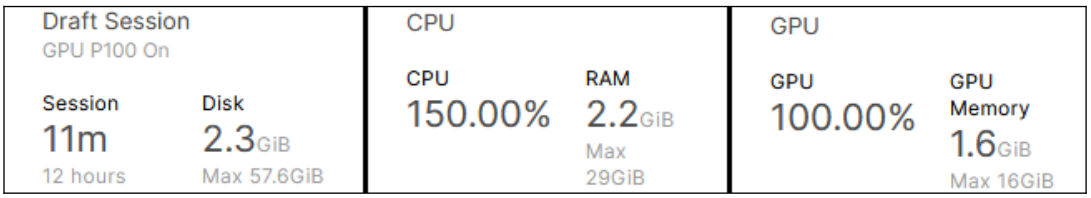

Figure 7: System resource utilization during model training on NVIDIA Tesla P100 GPU

6. Cybersecurity and Biometric Implications

6.1. Integration in Biometric Authentication Systems

The capacity to generate high-fidelity synthetic fingerprints raises important considerations for both security researchers and practitioners. On one hand, it offers a privacy-preserving means of advancing biometric systems by enabling robust testing and algorithmic development without exposing sensitive personal information. On the other hand, it introduces potential vulnerabilities that adversaries could exploit if protective measures are not effectively enforced [20, 21].

By accurately capturing the visual and structural attributes of real fingerprints, the proposed generative model can supply large synthetic datasets for training and validation in biometric authentication systems. These "privacy-friendly" samples minimize the legal and ethical constraints associated with collecting user data at scale, while still reflecting realistic ridge patterns crucial for ensuring system reliability.

In practice, developers can use this influx of synthetic samples to enhance feature extraction methods, improve fingerprint-matching algorithms, and conduct comprehensive stress testing against external conditions such as image quality variations or sensor discrepancies. The resulting improvements in accuracy and robustness can bolster consumer confidence in biometric authentication solutions across various sectors, including mobile devices, secure facility access, and e-Government initiatives [22].

6.2. Adversarial Vulnerabilities and Mitigation

While the generation of realistic fingerprint images aids legitimate research, it concurrently highlights potential avenues for adversarial attacks. Malicious actors could use convincing synthetic

fingerprints to probe or bypass fingerprint recognition systems. This possibility underscores the need to develop spoof detection or presentation attack detection mechanisms capable of distinguishing artificially generated ridges from authentic biometric inputs.

Countermeasures may include specialized classifiers trained on adversarially generated images, advanced liveness detection technologies, or multi-factor authentication procedures that combine fingerprint data with other identifiers. By integrating these preventative strategies, security experts can leverage the benefits of synthetic biometric training data without compromising user safety and privacy [23, 24, 25, 26, 27, 28, 29].

7. Conclusion and Future Directions

High-fidelity synthetic fingerprints offer both opportunities for biometric advancement and risks of misuse. They enable privacy-preserving testing and development but also raise security concerns if safeguards are lacking.

While the proposed EADC-GAN framework outperforms traditional DCGANs in terms of convergence speed and visual fidelity, it still requires substantial GPU resources for optimal performance. Moreover, the current evaluation relies primarily on qualitative assessments, lacking explicit quantitative metrics such as the Fréchet Inception Distance or specialized fingerprint matching scores. Additionally, the model's ability to generate ultra-high-resolution fingerprints (e.g., >256×256 pixels) remains to be fully explored — a critical requirement for certain forensic or high-security applications.

Future work may focus on scaling the architecture to support higher resolutions and embedding domain-specific fingerprint features tailored for forensic or advanced biometric scenarios [30–32]. Exploring alternative normalization strategies, such as hybrid or adaptive instance normalization [33-35], could further enhance ridge style consistency. Incorporating quantitative evaluation metrics specific to fingerprint quality — potentially benchmarked against live-capture datasets — would also strengthen the model's practical utility.

Finally, integrating spoof-detection modules directly into the training loop may help preemptively mitigate adversarial vulnerabilities [36-39]. By pursuing these directions, the proposed framework can evolve into a robust and versatile tool for both secure biometric authentication [40, 41] and adversarial AI research [42, 43].

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," Pattern Recognition Letters, vol. 79, pp. 80-105, 2016.

[2] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometric Recognition Using Deep Learning: A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3543-3567, 2022.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, J. Bengio, "Generative Adversarial Networks," in *Proceedings of the International Conference on Neural Information Processing Systems* (NIPS) 2014, pp. 2672—2680.

[4] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.

[5] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 4401-4410.

[6] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223-2232.

[7] O. Striuk and Y. Kondratenko, "Adaptive Deep Convolutional GAN for Fingerprint Sample Synthesis," in *Proceedings of 2021 IEEE 4th International Conference on Advanced Information and Communication Technologies* (AICT), 2021, pp. 193—196.
[8] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 214-223.
[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved Training of Wasserstein GANs," in Advances in Neural Information Processing Systems, 2017, pp. 5767-5777.
[10] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, "Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect," arXiv preprint arXiv:1803.01541, 2018.
[11] S. Yoon, J. Feng, and A. K. Jain, "Latent fingerprint enhancement via robust orientation field estimation," IEEE Transactions on Information Forensics and Security, vol. 10, no. 4, pp. 877-887, 2015.
[12] R. V. S. Srikanth, A. N. Kumar, "Synthetic Fingerprint Generation using Generative Adversarial Network," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 962-967.
[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International conference on machine learning, 2015, pp. 448-456.
[14] O. S. Striuk, Y. P. Kondratenko, "Optimization Strategy for Generative Adversarial Networks Design," International Journal of Computing, vol. 22, issue 3, pp. 292-301, 2023.
[15] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 1510-1519.
[16] E. S. Lubana, R. P. Dick, and H. Tanaka, "Beyond BatchNorm: Towards a unified understanding of normalization in deep learning," arXiv preprint arXiv:2106.05956, 2021.
[17] Y. Jing et al., "Neural Style Transfer: A Review" in IEEE Transactions on Visualization & Computer Graphics, vol. 26, no. 11, pp. 3365-3385, Nov. 2020.
[18] Y. I. Shehu, A. Ruiz-Garcia, V. Palade, A. James, "Sokoto Coventry Fingerprint Dataset," arXiv preprint arXiv:1807.10609, 2018.
[19] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization," in 3rd International Conference for Learning Representations, arXiv preprint arXiv:1412.6980, 2015.
[20] H. Kim, X. Cui, M. -G. Kim and T. H. B. Nguyen, "Fingerprint Generation and Presentation Attack Detection using Deep Neural Networks," in *Proceedings of 2019 IEEE Conference on Multimedia Information Processing and Retrieval* (MIPR), 2019, pp. 375—378, https://doi.org/10.1109/MIPR.2019.00074.
[21] Striuk, O.S., Kondratenko, Y.P. (2023). Generative Adversarial Networks in Cybersecurity: Analysis and Response. In: Kondratenko, Y.P., Kreinovich, V., Pedrycz, W., Chikrii, A., Gil-Lafuente, A.M. (eds) Artificial Intelligence in Control and Decision-making Systems. Studies in Computational Intelligence, vol 1087. Springer, Cham. https://doi.org/10.1007/978-3-031-25759-9_18
[22] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: A new representation and matching technique for fingerprint recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 12, pp. 2128-2141, 2010.
[23] C. Sousedik and C. Busch, "Presentation attack detection methods for fingerprint recognition systems: a survey," IET Biometrics, vol. 3, no. 4, pp. 219-233, 2014.
[24] A. Roy, N. Memon, J. Togelius and A. Ross, "Evolutionary Methods for Generating Synthetic MasterPrint Templates: Dictionary Attack in Fingerprint Recognition," 2018 International Conference on Biometrics (ICB), Gold Coast, QLD, Australia, 2018, pp. 39-46, https://doi.org/10.1109/ICB2018.2018.00017.
[25] S. A. Grosz, K. P. Wijewardena, and A. K. Jain, "ViT Unified: Joint Fingerprint Recognition and Presentation Attack Detection," arXiv preprint arXiv:2305.07602, 2023.
[26] S. Purnapatra et al., "Presentation Attack Detection with Advanced CNN Models for Noncontact-based Fingerprint Systems," arXiv preprint arXiv:2303.05459, 2023.
[27] A. Rai et al., "An Open Patch Generator based Fingerprint Presentation Attack Detection using Generative Adversarial Network," arXiv preprint arXiv:2306.03577, 2023.
[28] O. Striuk, Y. Kondratenko, I. Sidenko and A. Vorobyova, "Generative Adversarial Neural Network for Creating Photorealistic Images," *2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT)*, Kyiv, Ukraine, 2020, pp. 368-371, https://doi.org/10.1109/ATIT50783.2020.9349326.

[29] O. Striuk and Y. Kondratenko, “Cross-Domain Reconfigurable GAN with Fuzzy Components for Anomaly Detection,” 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 2023, pp. 1-5, https://doi.org/10.1109/DESSERT61349.2023.10416521.
[30] O.S. Striuk, Y.P. Kondratenko, “Generative Adversarial Neural Networks and Deep Learning: Successful Cases and Advanced Approaches,” *International Journal of Computing*, vol. 20, issue 3, pp. 339-349, 2021.
[31] O. Striuk and Y. Kondratenko, “Implementation of Generative Adversarial Networks in Mobile Applications for Image Data Enhancement,” *Journal of Mobile Multimedia*, vol. 19, no. 03, pp. 823–838, 2023. https://doi.org/10.13052/jmm1550-4646.1938.
[32] Y. Kondratenko *et al.*, “Analysis of the Priorities and Perspectives in Artificial Intelligence Implementation,” *2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Athens, Greece, 2023, pp. 1-8, https://doi.org/10.1109/DESSERT61349.2023.10416432.
[33] Z. Zhuo et al., “HybridNorm: Towards Stable and Efficient Transformer Training via Hybrid Normalization,” arXiv preprint arXiv:2503.04598, 2025.
[34] J. Zhu et al., “Transformers without Normalization,” arXiv preprint arXiv:2503.10622, 2025.
[35] B. Faye, H. Azzag, M. Lebbah, “Cluster-Based Normalization Layer for Neural Networks,” arXiv preprint arXiv:2403.16798, 2024.
[36] B. Adami, N. Karimian, “GRU-AUNet: A Domain Adaptation Framework for Contactless Fingerprint Presentation Attack Detection,” arXiv preprint arXiv:2504.01213, 2025.
[37] B. Adami et al., “A Universal Anti-Spoofing Approach for Contactless Fingerprint Biometric Systems,” arXiv preprint arXiv:2310.15044, 2023.
[38] A. Vurity et al., “ColFigPhotoAttnNet: Reliable Finger Photo Presentation Attack Detection Leveraging Window-Attention on Color Spaces,” arXiv preprint arXiv:2503.05247, 2025.
[39] M. F. Ramos et al., “Secure Multi-Party Biometric Verification using QKD assisted Quantum Oblivious Transfer,” arXiv preprint arXiv:2501.05327, 2025.
[40] E. Kablo et al., “The (Un)suitability of Passwords and Password Managers in Virtual Reality,” arXiv preprint arXiv:2503.18550, 2025.
[41] S. Cavasin et al., “Fingerprint Membership and Identity Inference Against Generative Adversarial Networks,” arXiv preprint arXiv:2406.15253, 2024.
[42] H. Fisher, M. Shahar, Y. S. Resheff, “Neural Fingerprints for Adversarial Attack Detection,” arXiv preprint arXiv:2411.04533, 2024.
[43] H. Kaur, R. Shukla, I. Echizen, P. Khanna, “Secure and Privacy Preserving Proxy Biometrics Identities,” arXiv preprint arXiv:2212.10812, 2022.