https://doi.org/10.62836/iaet.2021.1003

Global Science Publishing

IAET

Innovations in Applied Engineering and Technology

https://ojs.sgsci.org/journals/iaet

Article

Distributed Data Parallel Acceleration-Based Generative Adversarial Network for Fingerprint Generation

Shuguang Xiong [1], Huitao Zhang [2,*], Meng Wang [3] and Ning Zhou [4]

[1] *Baidu Inc. Beijing 100085, China; xiongshuguang@baidu.com*

[2] *Bybit Global Digital Solutions FZE, Dubai, United Arab Emirates*

[3] *Newmark Group, New York City, NY, USA; wang070210@gmail.com*

[4] *ByteDance Technology Co., Ltd., ShenZhen 518000, China; zhouning723@gmail.com*

Abstract: In the expanding landscape of artificial intelligence, scaling model training to accommodate larger and more intricate neural networks and datasets is imperative. This study addresses the scaling issue by employing Distributed Data Parallel (DDP) frameworks to enhance the training of deep learning models, specifically focusing on the generation of synthetic fingerprints. Utilizing DDP enables efficient management of vast datasets essential for training generative models, ensuring comprehensive coverage of the variability inherent in fingerprints. Moreover, the application of DDP in fingerprint generation not only expedites the training process but also enhances data security by distributing computation across multiple nodes. The effectiveness of DDP is demonstrated through substantial improvements in training efficiency, as evidenced by reduced training times and balanced Graphics Processing Unit (GPU) utilization rates. However, the study reveals challenges in GPU underutilization with larger batch sizes, indicating opportunities for optimizing resource allocation. Advances in Deep Convolutional Generative Adversarial Network (DCGAN) architecture is also discussed, highlighting the model's capability to create realistic synthetic fingerprints and suggesting a future focus on algorithmic adaptability and network sophistication.

Keywords: component; distributed learning; GAN; fingerprint generation

1. Introduction

In today's world of artificial intelligence and machine learning, the ability to scale up model training is increasingly important as both the models and the data they use get larger and more complex [1–3]. Distributed model training has become essential for dealing with these challenges. Artificial Intelligence (AI) models, especially those in the deep learning realm, have evolved to become more complex, with larger architectures and increased data requirements [4–6]. This development is motivated by advances in the biological domain, which has seen substantial progress in recent years [7–13]. These models now have billions of parameters and are trained on huge datasets. Traditional training methods that rely on a single computer are no longer sufficient. Distributed model training has become crucial for effective learning. It splits the training workload across several computers, cutting down the time needed to train models and making it possible to manage data volumes that a single machine couldn't handle alone.

Received: 15 November 2021; Accepted: 17 December 2021.

* Corresponding: Huitao Zhang (lucker0087@gmail.com)

Among the various strategies for distributed training, Distributed Data Parallel (DDP) stands out as a robust framework designed to optimize and accelerate the training process [14]. DDP works by replicating the neural network across multiple GPUs or nodes, each of which processes a subset of the data. Gradients from each node are then aggregated to update the models synchronously, ensuring consistency and improving the speed of convergence. This method not only enhances the training efficiency but also scales effectively as more computational resources are added, making it highly suitable for extensive neural network applications.

Fingerprint recognition is a biometric method used to identify individuals based on the unique patterns of ridges and valleys on their fingertips [15,16]. This technique has been in use for over a century, and it remains one of the most widely recognized and reliable forms of biometric identification. The importance of fingerprint recognition lies in its uniqueness; no two individuals, even twins, have identical fingerprints, making it an incredibly secure and accurate method of verifying identity. It is extensively used in various security applications, ranging from law enforcement and forensic science to consumer electronics like smartphones and access control systems. The ability to quickly and accurately verify a person's identity using their fingerprints makes it indispensable for enhancing security and preventing fraud in both the public and private sectors.

In this field, there are two primary applications: Fingerprint Verification (FPV) and Fingerprint Identification. FPV involves one-to-one matching, where a specific fingerprint is checked to confirm if it matches the one that has been previously registered as authentic. For instance, in a system accessible to persons A, B, C, and D, each user would be assigned a unique ID. If person A wants to access the system, his or her fingerprint would be compared only to the fingerprint linked to person A's ID. This process is known as FPV. On the other hand, in fingerprint identification, individual IDs are not assigned. Instead, anyone attempting to access the system would have their fingerprint compared against the entire database of authenticated fingerprints, which includes those of persons A, B, C, and D. This method is used for identifying the user from a group of authorized individuals.

Over the past decade, Convolutional Neural Networks (CNNs) have significantly advanced various computer vision and pattern recognition tasks, including Maritime image analysis. In [17], researchers employed a patch-based Siamese CNN alongside Gradient-weighted Class Activation Mapping (Grad-CAM) to demonstrate how CNNs can autonomously learn to recognize the critical minutiae points essential for fingerprint matching. Furthermore, in [18], a specific CNN called FingerNet enhanced processing techniques and achieved superior matching accuracy. Additionally, in [19], a new CNN known as minutiaeNet was introduced, integrating two sub-networks, CoarseNet and FineNet, to improve performance. This integration led minutiaeNet to become one of the leading algorithms in terms of precision and recall. However, training in a deep learning-based fingerprint recognition model requires a lot of images. The number of fingerprints in the current databases is very limited. In this case, generation of fingerprints can be considered as a possible solution. Fingerprint generation is a field where AI has had a transformative impact, offering new ways to create biometric models that are both unique and difficult to replicate. Traditional methods of fingerprint analysis and synthesis often require intensive computational efforts and are limited by the quality and quantity of fingerprint data available. However, with the implementation of AI-driven techniques, particularly through the use of generative adversarial networks (GANs), the process of generating high-quality fingerprint images can be significantly improved. GANs involve a dual-network architecture where one network generates candidates and the other evaluates them, promoting a continuous enhancement in the quality of generated fingerprints.

The application of distributed training methods like DDP to fingerprint generation has several compelling advantages. First, it allows for the handling of datasets of fingerprint images, which is crucial for training robust generative models. The ability to process large volumes of data ensures that the diversity and variability in fingerprints are adequately captured, leading to models that can generalize well across different populations and scenarios. Additionally, distributed training facilitates quicker iteration cycles, enabling researchers and practitioners to refine models more rapidly and deploy them in practical applications, such as secure biometric authentication systems. Moreover, in the context of security and privacy, distributed training can also provide an added layer of data protection. By decentralizing the data processing, sensitive information can be handled locally, reducing the risk of data breaches that could occur if all data were processed on a single server.

Incorporating model compression techniques can be pivotal in this context, which this article aims to explore further. Model compression not only reduces the computational demands and storage requirements of deploying deep learning models on resource-constrained devices, it also enhances the efficiency of processing large datasets typical of the telecom industry. By applying these techniques, telecom companies can leverage advanced data analytics for customer segmentation and retention strategies more effectively, enabling faster and more cost-efficient decision-making processes that are crucial for maintaining competitive advantage in a rapidly evolving market.

This paper is structured as follows: section 2 details the related works of fingerprint recognition and generation. Subsequently, section 3 provides the workflow of the proposed method. The experimental results and corresponding discussion are provided in section 4. Finally, section 5 provides a comprehensive conclusion of this paper.

2. Literature Review

2.1. Fingerprint recognition

Fingerprint recognition methods fall into two main categories: traditional image processing and modern deep learning algorithms. Traditional image processing techniques have seen numerous developments over the past decades. For example, Kaur et al. introduced a fingerprint matching system that includes a minutiae extractor and matcher [20]. This method follows several specific steps: data preprocessing, minutiae extraction and matching, and data post-processing, with experimental results affirming its effectiveness. Khan et al. developed a robust matching technique that uses multiple thresholds to improve image quality before binarization and minutiae extraction using a 3x3 window [21]. Jea et al. introduced a method that utilizes localized secondary features from relative minutiae data, employing a flow network-based technique for matching these features effectively [22]. Tan et al. introduced a fingerprint scheme based on the genetic algorithm that showed good performance on the NIST-4 fingerprint dataset [23]. Raja et al. presented a fingerprint recognition method called Fingerprint Recognition based on Minutiae Score Matching (FRMSM), which involves a block filter for fingerprint thinning to maintain image quality and extract detailed features [24].

Despite their successes, traditional fingerprint matching techniques face several challenges that limit their practical application. Firstly, they are highly sensitive to the quality of the fingerprint image, where poor-quality images often lead to inaccurate or false matches. Secondly, they struggle with variations in fingerprint images caused by changes in orientation, pressure, or moisture content on the finger, making them less adaptable to different scenarios. Thirdly, traditional methods are not suited for large-scale applications, such as biometric authentication systems that need to match fingerprints against extensive databases. Fourthly, these methods are vulnerable to spoofing attacks, where attackers attempt to deceive the system with fake fingerprints. Finally, traditional approaches often require specialized hardware like high-resolution fingerprint scanners, which are expensive and can limit system scalability.

Given the limitations of traditional fingerprint recognition methods, deep learning algorithms have emerged as a promising alternative in recent years. Deep learning focuses on developing algorithms inspired by the structure and functionality of the human brain, notably through models such as Convolutional Neural Networks (CNNs), which are widely used in various computer vision tasks including biometric recognition.

Recent research in fingerprint recognition has explored the potential of CNNs. For instance, Zeng et al. developed a residual network to extract local features from fingerprint images, enhancing the structure of the CNN. This network uses Cross-Entropy and Contrast-Loss functions for training and employs the K-means++ algorithm to stabilize feature extraction, ultimately improving recognition performance for certain fingerprints [25]. Deshpande et al. introduced an efficient fingerprint authentication model using a deep CNN with fifteen layers, structured in two phases: the preparation stage, which includes image collection, augmentation, and pre-processing, and the feature extraction and matching stage, leading to optimal matching performance [26].

Su et al. adopted a deep learning method for pore extraction on fingerprints, utilizing CNNs for feature learning and classification. Their study also introduces a novel affine Fourier moment-matching (AFMM) technique that addresses both local and global linear distortions by matching and fusing scores from different fingerprint features [27]. Additionally, Wu et al. proposed a method for recognizing fouled and damaged fingerprints using a CNN-based algorithm called Central Block Fingerprint and Fuzzy Feature Points Fingerprint (CBF-FFPF). This method combines sub-blocks centered on the fingerprint core from thinned images with a fuzzy graph of feature points to achieve effective recognition rates [28].

While these deep learning approaches have shown promise in fingerprint identification, they still require further refinement to match the accuracy of traditional minutiae point matching. A significant challenge is the lack of sufficient data, as privacy concerns make it difficult to collect fingerprints compared to other biometric data like facial images. Enhancing the volume of data available is critical, as it would allow deep learning models to capture more feature information and improve their predictive capabilities.

2.2. Fingerprint generation

In the realm of fingerprint generation, deep learning models have shown potential, similar to their applications in fingerprint matching. For example, Fahim et al. used a combination of residual networks and spectral normalization to create fingerprints [29]. Their method, featuring average residual connections, was more effective in preventing vanishing gradients compared to traditional residual connections. Spectral normalization helped stabilize weight variation within the network. They also used the Multi-scale Structural Similarity (MS-SSIM) metric to evaluate the diversity of the generated samples, indicating that their approach could produce a varied set of images while reducing the risk of mode collapse.

Minaee et al. developed a machine learning framework based on Generative Adversarial Networks (GANs), enhancing it with a specific regularization term in the loss function. The model's performance was measured using the Frechet Inception Distance (FID), showing significant quantitative improvements [30].

3. Method

3.1. Dataset Preparation

The FVC2002 and FVC2004 competitions are notable events in the field of fingerprint matching. Across these two competitions, a total of eight fingerprint databases were compiled. Each database contains 800 fingerprint images from 100 different individuals, providing a substantial resource for analysis and study. It's important to highlight that this research specifically utilizes the genuine fingerprint databases: FVC 2002 DB1, FVC 2002 DB2, FVC 2002 DB3, FVC 2004 DB1, FVC 2004 DB2, and FVC 2004 DB3. The synthetic databases, FVC 2002 DB4 and FVC 2004 DB4, are excluded from this study. Consequently, the datasets employed here comprise 4800 fingerprint images across the six mentioned databases. Figure 1 in the document displays sample images from these collected datasets.

Figure 1. The sample images of the collected datasets.

3.2. Deep Convolutional Generative Adversarial Network (DCGAN)-Based Fingerprint Generation

GANs, or Generative Adversarial Networks [31,32], are a type of deep learning model that creates synthetic data similar to an existing dataset. These networks consist of two parts: the generator and the discriminator, as illustrated in Figure 2. The generator starts with random noise to produce new data samples. The discriminator then evaluates these samples alongside real data, determining whether each piece of data is real or fake. During training, the generator tries to fool the discriminator into accepting the synthetic data as real. At the same time, the discriminator works to tell the real data from the fake more accurately. The goal for the generator is to get better at making data that looks like the real thing, while the discriminator aims to get better at spotting the fakes. This competition drives the generator to create more and more convincing data over time. Eventually, GANs can generate new data that closely mimics the original, useful for tasks like creating images, composing music, and processing language.

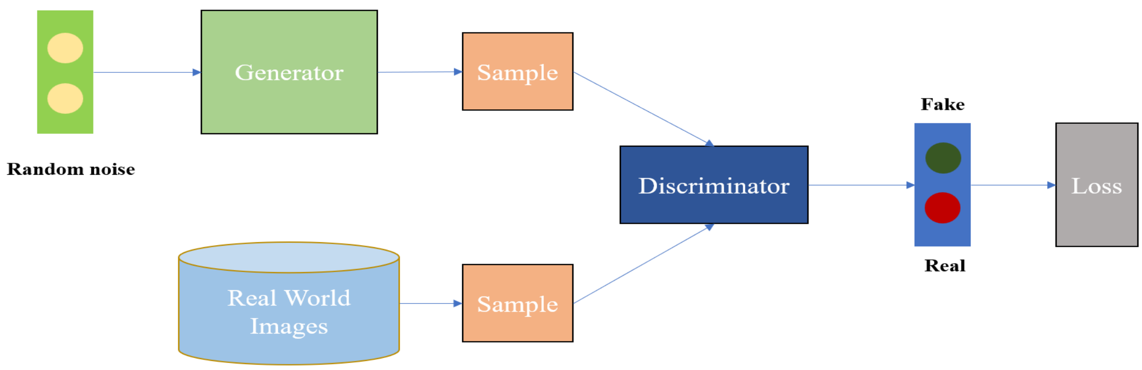

Figure 2. The workflow of GAN.

GANs are well-known for their capability to produce high-quality, varied, and intricate data. They are used in many applications such as creating art, enhancing datasets, and transferring styles. However, traditional GAN models often face a problem known as mode collapse during training. This issue can cause the generation of images to fail. Additionally, GANs come with other challenges, such as unstable training, sensitivity to changes in settings (hyperparameters), and unreliable outcomes in terms of convergence. However, traditional GANs often face several issues, such as training instability, mode collapse (where the generator produces limited varieties of outputs), and difficulties in convergence. DCGAN, or Deep Convolutional Generative Adversarial Network, is a variant of the basic GAN that primarily incorporates convolutional and convolutional-transpose layers in the generator and discriminator, respectively. This structure is inspired by the deep convolutional networks commonly used in image analysis, which makes DCGAN particularly effective for tasks involving images. DCGAN addresses these problems by incorporating convolutional layers in both the generator and discriminator, which helps in stabilizing the training process. It also uses techniques like batch normalization and leaky ReLU activations to ensure a smoother gradient flow and more stable distribution of activations throughout the network. These modifications make DCGAN more robust and capable of generating higher quality images, while significantly reducing the risk of mode collapse and improving the overall training dynamics.

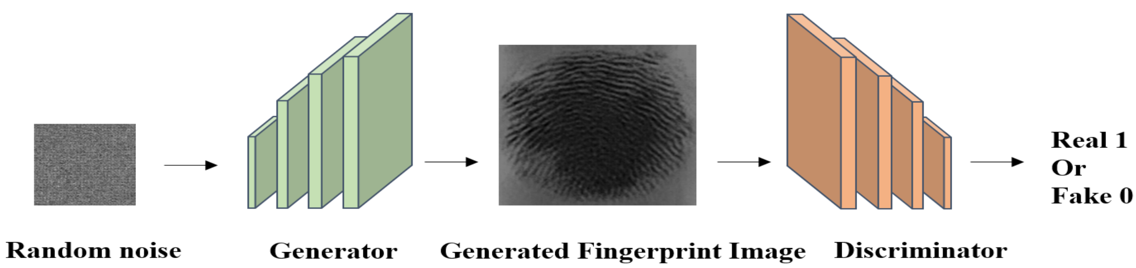

Figure 3. The workflow of DCGAN for fingerprint generation.

In this study, the proposed DCGAN model shown in Figure 3 consists of two main components: a generator and a discriminator. The generator begins with a 200-dimensional vector drawn from a random normal distribution. This vector is first expanded through a fully connected layer into a 65, 536-dimensional vector. After this expansion, the vector passes through a batch normalization layer and is activated using a leaky ReLU function. The result is then reshaped into a 16×16×256 structure. This structure is processed through several two-dimensional deconvolution layers and batch normalization layers, which are applied in a sequence. During this sequence, the structure undergoes two iterations where the number of channels is reduced sequentially (64, then 32). A final deconvolution layer then transforms this output into a 256×256×1 image that matches the dimensions of the original input image.

For the discriminator, the process starts with the generated image being processed by two convolutional layers, each followed by a ReLU activation function. The output from these layers is flattened into a one-dimensional vector, leading to a final prediction that classifies the image as either true or false.

3.3. Distributed Data Parallel

This study uses the DDP to enhance the efficiency and scalability of training DCGAN models across multiple computing devices. DDP operates by replicating the model across multiple GPUs where each device contains a complete copy of the model. During training, the dataset is divided into smaller batches, and each batch is processed by a different GPU. This distribution allows simultaneous computations, vastly improving training speed and efficiency. One of the key features of DDP is the synchronization of gradients across all devices. After each forward and backward pass, DDP aggregates the gradients from all GPUs to update the model parameters consistently. This synchronization ensures that each model replica converges towards the same solution, maintaining model accuracy and integrity despite the distributed nature of the training.

DDP utilizes efficient communication strategies, such as ring-reduce or all-reduce algorithms, to minimize communication overhead between GPUs. These algorithms optimize the way data is transferred across the network, reducing the time spent on data exchanges and maximizing the computational work performed during training. The choice of algorithm can depend on the specific network architecture and the number of GPUs involved, aiming to achieve the best possible performance.

3.4. Implementation Details

In this research, the configuration of the DCGAN model involves carefully selected parameters to optimize performance and ensure effective learning. The choice of Binary Crossentropy [33] as the loss function is strategic for handling the binary classification nature of the discriminator's output in determining whether images are real or generated. We have set the training to run for 3000 epochs, allowing the model ample time to converge and learn detailed features from the data. The optimizer used is Adam, known for its efficiency in handling sparse gradients and its adaptive learning rate capabilities, which is set at a very low value of 0.00001 to prevent overshooting during training updates. Additionally, the entire training process is accelerated using multiple RTX 3080 GPUs. This hardware choice not only speeds up the computations necessary for training the deep learning model but also supports the extensive data handling required by the large networks of the DCGAN, especially given the high-dimensional transformations involved in the generator and discriminator.

4. Results and Discussion

4.1. The Performance of the Model

Figure 4 represents the outputs of a DCGAN over time as it learns to generate fingerprint images. Initially, at epoch 0, we see a uniform gray square where the model begins without any learned features or patterns. This image signifies the starting point of the model's knowledge—essentially a blank canvas from which it must learn to create a complex structure.

By epoch 10, the network began to understand the fundamental components of fingerprint patterns. We can see the emergence of curving lines and ridges that are characteristic of a fingerprint. However, the image is still blurry and lacks definition. The patterns are somewhat repetitive and lack the uniqueness and clarity of genuine fingerprints. At this stage, the generator is starting to apply the foundational knowledge it has learned about fingerprint structures, while the discriminator is learning to differentiate these early attempts from real fingerprint images.

Fast forward to epoch 500, and the transformation is remarkable. The image now shows a well-defined and clear fingerprint, with the intricate loops and whorls that we would expect to find in a real print. The ridges and valleys are distinct, and the image has a realistic texture and depth. This significant improvement is the result of the adversarial process, where the generator and discriminator have been in a constant tug-of-war, refining their abilities with each iteration. The generator has become adept at crafting images that mimic the intricacies of real fingerprints, and the discriminator has pushed it to this level of proficiency by getting better at identifying fake prints.

This progression demonstrates the DCGAN's powerful capability to generate realistic synthetic data. Not only does this have implications for technology that relies on biometric security, but it also shows potential for use in forensic science, where high-quality fingerprint images are necessary for identification and analysis. More examples of generated fingerprint images using DCGAN model are provided in Figure 5.

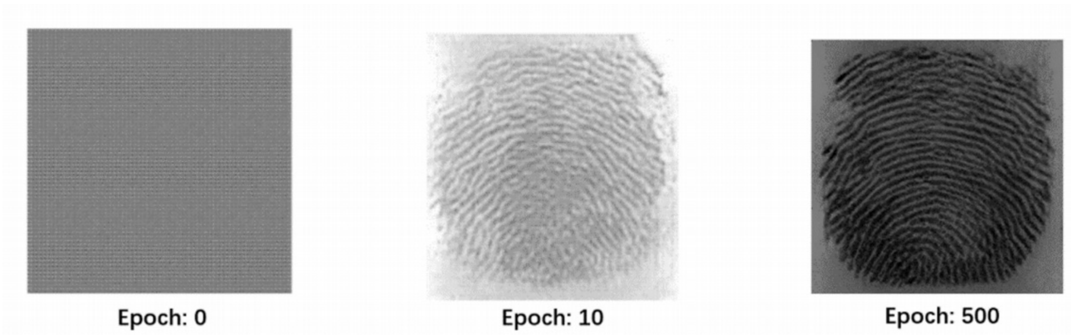

Figure 4. The visualization of generated fingerprint images during the training process based on the DCGAN model.

Figure 5. More examples of generated fingerprint images using DCGAN model.

Table 1 presents for the DCGAN model, which does not use data parallelism, the table shows a training time of 17 minutes and a GPU utilization rate of 67%. In contrast, the DDP-based DCGAN model, which employs two GPUs for data parallelism, has a reduced training time of 10 minutes. The GPU utilization rates for the DDP-based model are listed as 54% and 51%, which likely correspond to the utilization rates for each of the two GPUs used. The information suggests that by using DDP to distribute the workload across two GPUs, there is a significant improvement in training efficiency, reducing the time required by almost half. However, it also shows a lower utilization rate for each GPU, which is a common occurrence in distributed computing scenarios where the workload is spread across multiple processors.

Table 1. Comparison of DCGAN and DDP-based DCGAN model.

Model (data parallel)	Time (min)	GPU Utilization rate
DCGAN	17	67%
DDP-based DCGAN (two GPUs)	10	54%,51%

Figure 6 illustrates the relationship between batch size and GPU utilization during the training of a deep learning model. The x-axis represents the batch size, with values ranging from 8 to 256, while the y-axis denotes the GPU utilization percentage, starting from 67% and gently declining to 51% as the batch size increases. This diminishing utilization rate with larger batches could imply that the GPU has more processing power than what is required for larger batches, leading to underutilization. This information can be critical for optimizing GPU performance and ensuring efficient use of computational resources during model training. The optimal batch size for maximum GPU efficiency appears to be between 8 and 32, beyond which the benefits of increased batch size do not proportionately leverage the available GPU power. The decrease in GPU utilization with larger batch sizes is likely due to how GPUs handle parallel processing. GPUs excel when they can perform many tasks at once, which happens with smaller batch sizes. However, as batch sizes get larger, each task can take up more GPU resources, leading to a situation where there's not enough work for all GPU cores at once or memory bandwidth becomes a bottleneck. This means the GPU isn't used to its full potential, and utilization goes down.

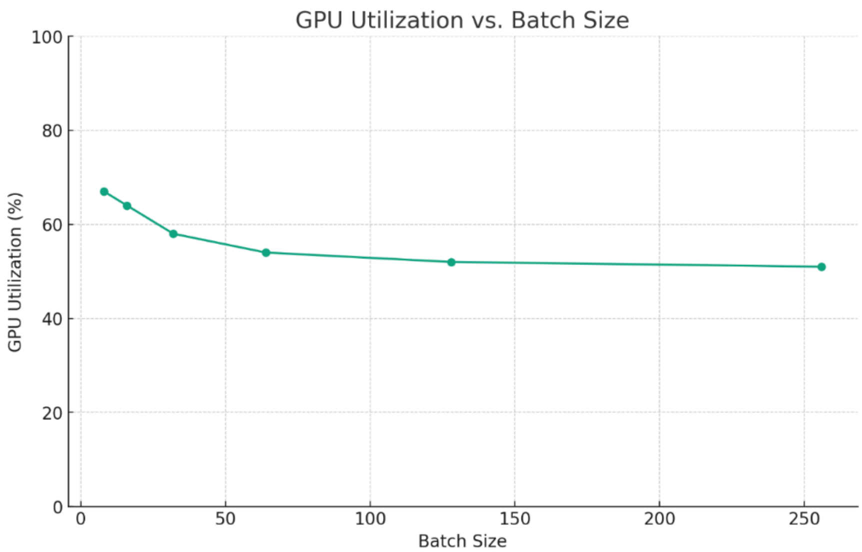

Figure 6. The correlation of batch size and GPU utilization.

4.2. Discussion

The results we are seeing with DCGANs are definitely encouraging, but we have still got some work to implement. Even though the fingerprints it's creating look more and more like the real thing, they're not quite perfect yet. They're missing those little quirks that make real fingerprints unique, and we need to get those right, especially for high-security uses where accuracy is everything using some possible advanced methods [34–37]. It should be also to think about the right way to use this technology to make sure it's not misused. On the tech side of things, using DDP to train these networks is making things a lot quicker, especially with two GPUs. It's great at sharing the work and speeding up the process. But when we use bigger batches of data, the GPUs aren't working as hard as they could be, which means we might not be using our resources the best way we can. It's like we need smarter systems that can figure out the best way to keep the GPUs busy combined with some advanced hardware systems [38–44].

Looking ahead, we need to get better at finding the sweet spot between the size of the data batches and how much work the GPUs are doing. Also, if we can make DDP smarter about how it hands out the work, we could train even more complex networks without wasting time or power. As we keep improving how DCGANs are built and trained, we might try out new structures for the networks or mix different types together to get the best results. And imagine if DDP could learn on the fly how to handle different amounts of work—that would really shake things up. Getting these things right won't just make the training process better; it could make DCGANs useful for all kinds of new jobs that need synthetic data.

5. Conclusion

The findings from this study highlight the vital role that distributed training frameworks play in enhancing biometric security measures, like the creation of artificial fingerprints using the proposed DCGAN. It can be found that using DDP not only cuts down on the time it takes to train models but also paves the way for refining and applying these models more effectively. However, we have hit a stumbling block with how GPUs are used less as we crank up the amount of data they work with. Looking ahead, it is clear we need to tweak these training settings to get the most out of our GPUs. The exciting part is thinking about the future—if DDP could automatically adjust to the amount of data it's given, it would be a game-changer for training the complicated networks we're working on. This could really push the boundaries of what we can do with DCGANs and the synthetic data they produce.

Funding

Not applicable.

Author Contributions

Conceptualization, S.X. and H.Z.; writ-ing—original draft preparation and writing—review and editing, S.X., H. Z., M.W. and N.Z. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

References

1 Luo Z, Xu H, Chen F. Audio Sentiment Analysis by Heterogeneous Signal Features Learned from Utterance-Based Parallel Neural Network. In Proceedings of the AffCon@ AAAI 2019, Honolulu, HA, USA, 27 January 2019.

2 Chen F, Luo Z, Xu Y, Ke D. Complementary Fusion of Multi-Features and Multi-Modalities in Sentiment Analysis. 2019. DOI: 10.48550/arXiv.1904.08138.

3 Luo Z, Zeng X, Bao Z, Xu M. Deep Learning-Based Strategy for Macromolecules Classification with Imbalanced Data from Cellular Electron Cryotomography. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019.

4 Qiu Y, Yang Y, Lin Z, Chen P, Luo Y, Huang W. Improved Denoising Autoencoder for Maritime Image

Denoising and Semantic Segmentation of USV. *China Communications* 2020; **17**(**3**): 46–57.

5 Sun G, Zhan T, Owusu BG, Daniel AM, Liu G, Jiang W. Revised Reinforcement Learning Based on Anchor Graph Hashing for Autonomous Cell Activation in Cloud-RANs. *Future Generation Computer Systems* 2020; **104**: 60–73.

6 Qiu Y, Chang CS, Yan JL, Ko L, Chang TS. Semantic Segmentation of Intracranial Hemorrhages in Head CT Scans. In Proceedings of the 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 18–20 October 2019.

7 Qiu Y. Estimation of Tail Risk Measures in Finance: Approaches to Extreme Value Mixture Modeling. *Ph. D. Thesis*, Johns Hopkins University, Baltimore, MD, USA, May 2019.

8 Wang B, Shen Y, Zhai L, Xia X, Gu H-M, Wang M, Zhao Y, Chang X, Alabi A, Xing S, Deng S, Liu B, Wang G, Qin S, Zhang D-W. W. Atherosclerosis-Associated Hepatic Secretion of VLDL but Not PCSK9 Is Dependent on Cargo Receptor Protein Surf4. *Journal of Lipid Research* 2021; **62**: 100091.

9 Deng SJ, Shen Y, Gu HM, Guo S, Wu SR, Zhang DW. The Role of the C-Terminal Domain of PCSK9 and SEC24 Isoforms in PCSK9 Secretion. *Biochimica et Biophysica Acta (BBA)-Molecular and Cell Biology of Lipids* 2020; **1865**(**6**): 158660.

10 Shen Y, Wang B, Deng S, Zhai L, Gu HM, Alabi A, Xia X, Zhao Y, Chang X, Qin S, Zhang DW. Surf4 Regulates Expression of Proprotein Convertase Subtilisin/Kexin Type 9 (PCSK9) but Is Not Required for PCSK9 Secretion in Cultured Human Hepatocytes. *Biochimica et Biophysica Acta (BBA) -Molecular and Cell Biology of Lipids* 2020; **1865**(**2**): 158555.

11 Xia D, Alexander AK, Isbell A, Zhang S, Ou J, Liu XM. Establishing a Co-Culture System for Clostridium Cellulovorans and Clostridium Aceticum for High Efficiency Biomass Transformation. *J Sci Heal Univ Ala* 2017; **14**: 8–13.

12 Horne J, Beddingfield E, Knapp M, Mitchell S, Crawford L, Mills SB, Wrist A, Zhang S, Summers RM. Caffeine and Theophylline Inhibit β-Galactosidase Activity and Reduce Expression in Escherichia Coli. *Acs Omega* 2020; **5**(**50**): 32250–32255.

13 Mock MB, Zhang S, Pniak B, Belt N, Witherspoon M, Summers RM. Substrate Promiscuity of the Ndmcde N7-Demethylase Enzyme Complex. *Biotechnology Notes* 2021; **2**: 18–25.

14 Li S, Zhao Y, Varma R, Salpekar O, Noordhuis P, Li T, Paszke A, Smith J, Vaughan B, Damania P, Chintala S. Pytorch Distributed: Experiences on Accelerating Data Parallel Training. *Proceedings of the VLDB Endowment* 2020; **13**(**12**): 3005–3018.

15 Maltoni D, Maio D, Jain AK, Prabhakar S. *Handbook of Fingerprint Recognition (Vol. 2)*; Springer: London, UK, 2009.

16 Marasco E, Ross A. A Survey on Antispoofing Schemes for Fingerprint Recognition Systems. *ACM Computing Surveys (CSUR)* 2014; **47**(**2**): 1–36.

17 Chowdhury A, Kirchgasser S, Uhl A, Ross A. Can a CNN Automatically Learn the Significance of Minutiae Points for Fingerprint Matching?. In Proceedings of the 2020 IEEE Winter Conference of Applications on Computer Vision (WACV 2020), 2–5 March 2020 Snowmass Village, CO, USA.

18 Tang Y, Gao F, Feng J, Liu Y. FingerNet: An Unified Deep Network for Fingerprint Minutiae Extraction. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017.

19 Nguyen DL, Cao K, Jain AK. Robust Minutiae Extractor: Integrating Deep Networks and Fingerprint Domain Knowledge. In Proceedings of the 2018 International Conference on Biometrics (ICB), Gold Coast, Australia, 20–23 February 2018.

20 Kaur M, Singh M, Girdhar A, Sandhu PS. Fingerprint Verification System Using Minutiae Extraction Technique. *International Journal of Computer and Information Engineering* 2008; **2**(**10**): 3405–3410.

21 Khan AI, Wani MA. Strategy to Extract Reliable Minutia Points for Fingerprint Recognition. In Proceedings of the 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, India, **21**–**22** February 2014.

22 Jea TY, Govindaraju V. A Minutia-Based Partial Fingerprint Recognition System. *Pattern Recognition* 2005;

38(**10**): 1672–1684.

23 Tan X, Bhanu B, Lin Y. Fingerprint Classification Based on Learned Features. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 2005; **35**(**3**), 287–300.

24 Raja KB. Fingerprint Recognition Using Minutia Score Matching. 2010. DOI: 10.48550/arXiv.1001.4186.

25 Zeng F, Hu S, Xiao K. Research on Partial Fingerprint Recognition Algorithm Based on Deep Learning. *Neural Computing and Applications* 2019; **31**(**9**): 4789–4798.

26 Deshpande UU, Malemath VS, Patil SM, Chaugule SV. CNNAI: A Convolution Neural Network-Based Latent Fingerprint Matching Using the Combination of Nearest Neighbor Arrangement Indexing. *Frontiers in Robotics and AI* 2020; **7**: 113.

27 Su HR, Chen KY, Wong WJ, Lai SH. A Deep Learning Approach Towards Pore Extraction for High-Resolution Fingerprint Recognition. In Proceedings of the 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017.

28 Wu Z, Wang Y, Zhang J. Fouling and Damaged Fingerprint Recognition Based on Deep Learning. *Journal of Electronics and Information (in Chinese)* 2017; **39**(**7**): 1585–1591.

29 Fahim MANI, Jung HY. A Lightweight GAN Network for Large Scale Fingerprint Generation. *IEEE Access* 2020; **8**: 92918–92928.

30 Minaee S, Abdolrashidi A. Finger-GAN: Generating Realistic Fingerprint Images Using Connectivity Imposed GAN. 2018. DOI: 10.48550/arXiv.1812.10482.

31 Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* 2018; **35**(**1**): 53–65.

32 Wang K, Gou C, Duan Y, Lin Y, Zheng X, Wang FY. Generative Adversarial Networks: Introduction and Outlook. *IEEE/CAA Journal of Automatica Sinica* 2017; **4**(**4**): 588–598.

33 Ruby U, Yendapalli V. Binary Cross Entropy with Deep Learning Technique for Image Classification. *Int J Adv Trends Comput Sci Eng* 2020; **9**(**10**). DOI: 10.30534/ijatcse/2020/175942020.

34 Luo Z, Xu H, Chen F. Utterance-Based Audio Sentiment Analysis Learned by a Parallel Combination of CNN and LSTM. ArXiv abs/1811.08065 (2018): n. pag.

35 Chen F, Luo Z. Learning Robust Heterogeneous Signal Features from Parallel Neural Network for Audio Sentiment Analysis. 2018. DOI: 10.48550/arXiv.1811.08065.

36 Chen F, Luo Z. Sentiment Analysis Using Deep Robust Complementary Fusion of Multi-Features and Multi-Modalities. *CoRR* 2019.

37 Liu Y, Bao Y. Review of Electromagnetic Waves-Based Distance Measurement Technologies for Remote Monitoring of Civil Engineering Structures. *Measurement* 2021; **176**: 109193.

38 Sugaya T, Deng X. Resonant Frequency Tuning of Terahertz Plasmonic Structures Based on Solid Immersion Method. In Proceedings of the 2019 44th International Conference on Infrared, Millimeter, And Terahertz Waves (IRMMW-THz), Paris, France, 1–6 September 2019.

39 Yu F, Milord J, Orton S, Flores L, Marra R. Students' Evaluation Toward Online Teaching Strategies for Engineering Courses during COVID. In Proceedings of the 2021 ASEE Midwest Section Conference, Virtual Conference, 13–15 September 2021.

40 Deng X, Li L, Enomoto M, Kawano Y. Continuously Frequency-Tuneable Plasmonic Structures for Terahertz Bio-Sensing and Spectroscopy. *Scientific Reports* 2019; **9**(**1**): 3498.

41 Deng X, Simanullang M, Kawano Y. Ge-Core/A-Si-Shell Nanowire-Based Field-Effect Transistor for Sensitive Terahertz Detection. *Photonics* 2018; **5**(**2**): 13.

42 Liu Y, Liu L, Yang L, Hao L, Bao Y. Measuring Distance Using Ultra-Wideband Radio Technology Enhanced by Extreme Gradient Boosting Decision Tree (Xgboost). *Automation In Construction* 2021; **126**: 103678.

43 Yu F, Strobel J. Work-in-Progress: Pre-college Teachers' Metaphorical Beliefs about Engineering. In Proceedings of the 2021 IEEE Global Engineering Education Conference (EDUCON), Vienna, Austria, 21–23 April 2021.

44 Milord J, Yu F, Orton S, Flores L, Marra R. Impact of COVID Transition to Remote Learning on

Engineering Self-Efficacy and Outcome Expectations. In Proceedings of the 2021 ASEE Virtual Annual Conference, Virtual Conference, 26–29 July 2021.