Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 3, June 2025, pp. 2370~2379
ISSN: 2302-9285, DOI: 10.11591/eei.v14i3.9190

Enhancing realism in handwritten text images with generative adversarial networks

Parul Dubey[1], Manjushree Nayak[2], Hitesh Gehani[3], Ashwini Kukade[4], Vinay Keswani[5], Pushkar Dubey[6]
[1]Department of Computer Science and Engineering, Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India
[2]Department of Computer Science and Engineering, NIST University, Berhampur, India
[3]School of Computer Science and Engineering, Ramdeobaba University, Nagpur, India
[4]Department of Artificial Intelligence, G H. Raisoni College of Engineering, Nagpur, India
[5]Department of Electronics and Telecommunication Engineering, G H. Raisoni College of Engineering, Nagpur, India
[6]Department of Management Pandit Sundarlal Sharma (Open) University Chhattisgarh, Bilaspur, India

Article Info

Article history:

Received Aug 23, 2024
Revised Dec 7, 2024
Accepted Dec 25, 2024

Keywords:

Discriminator
Generative adversarial network
Generator
Handwritten text
Image generation
Neural network

ABSTRACT

Image synthesis is particularly important for applications that want to create realistic handwritten documents, which is why handwritten text generation is a critical area within its domain. Even with today's highly advanced technology, generating diverse and accurate representations of human handwriting is still a tough problem because of the variability in style. In this study, we tackle the problem of instability during the training phase of generative adversarial networks (GANs) for generating handwritten text images. Using the MNIST dataset, which includes 60,000 training and 10,000 test images of handwritten digits, we trained a GAN model to generate synthetic handwritten images. The methodology involves optimizing both the generator and discriminator using adversarial training, binary cross-entropy loss, as well as the optimizer Adam. A brand-new decaying learning rate schedule was introduced to speed up convergence. Performance was evaluated using the Fréchet inception distance (FID) metric. The results show that this model effectively generated high-quality synthetic images of handwritten digits, which resembled real data closely in the face of it all and also that there was a steady reduction in FID scores across epochs indicating improved performance.

Corresponding Author:

Parul Dubey
Department of Computer Science and Engineering, Symbiosis Institute of Technology, Nagpur Campus
Symbiosis International (Deemed University)
Pune, India
Email: parul.dubey@sitnagpur.siu.edu

1. INTRODUCTION

There is a precedent in almost any realm of communication for the written word: in schools, at the inception of making, e.g., a novel, or in any legal documentation. The creation of accurate representations of handwritten text poses a number of challenges because of the complexity and variability of human handwriting. The specificities and oddities that characterise the numerous different forms of handwriting are numerous and impossible to replicate using hand-written text through traditional means. Goodfellow *et al.* [1] proposed generative adversarial networks (GANs) as one of the key concerns that might make commercial beans a solution to the problem in 2020. They use a generative model, which is a model that is able to learn to generate data samples that are not able to be distinguished from the original data.

Support vector machines (SVMs) have two neural networks as their basic architecture: the generator and the discriminator [2]. The generator should produce pictures the same as those available in the data set, and the discriminator should authenticate that the ideas generated are real, while the generator can even generate images from noise [3]. This makes training these two networks together an adversarial process. So essentially, the generator tries to outsmart the discriminator by generating more plausible-looking images, and the discriminator tries to do the same and not get fooled by the generator. This process produces visuals that are incredibly lifelike through a technique called adversarial training [4].

During this investigation, we took advantage of the ability of GANs to produce images of handwritten text. Our approach uses the MNIST dataset, which is commonly considered a touchstone for handwritten digit recognition, to train the GAN model. The discriminator network is simply getting trained to tell apart real MNIST images from fake images generated by the generator, and the generator is simply getting trained to generate images of handwritten numbers as close to those from the MNIST dataset as possible. Our primary job is to investigate the quality measures and realism of the generated handwritten text visuals using a wide range of metrics and quality assessments. This allows us to investigate how the performance of the GAN is impacted by using different training parameters and topologies. This research shows that GANs can generate high-quality handwritten text images and offers insights into the prospective uses of this technology. The details of the reviewed literature can be seen in Table 1. The primary problem addressed in this study is the instability inherent in training GANs, particularly when generating high-quality, realistic handwritten text images. This instability leads to fluctuating image quality and training inefficiencies, making it difficult to achieve consistent results, especially with varying hyperparameters like learning rates.

Table 1. Comparison of literature reviewed

Ref.	Authors	Year	Focus area	Key findings	Research gap
[5]	Karthika and Durgadevi	2021	Overview of GANs and applications.	Introduction to GANs, types, applications, limitations, future work.	No prior work adequately addresses stability issues in generating diverse and accurate handwritten text.
[6]	Gui *et al.*	2023	Comprehensive review of GAN algorithms.	Goals, mathematical representations, structures, theoretical questions, applications.	
[7]	Kundu *et al.*	2020	Text-line extraction from handwritten documents.	GAN-based TLE, U-Net and Patch GAN architectures, evaluation on HIT-MW and ICDAR 2013 datasets.	
[8]	Kang *et al.*	2020	Handwritten word image generation.	Method for realistic handwritten word images, calligraphic style features, textual content, few-shot setup.	
[9]	Shahriar	2022	GANs in art generation.	Survey on GAN-based art generation, visual arts, music, literary text, comparison of architectures.	
[10]	Lv *et al.*	2021	MRI reconstruction with GANs.	Comparative study on DAGAN, ReconGAN, RefineGAN, KIGAN, superior performance of RefineGAN.	

In this document, we summarise the whole paper as follows: section 2 details the method, which contains the GAN architecture, dataset, and training strategy. Section 3 presents both the results and analysis, highlighting the model performance along with the challenges in this field. Section 4 concludes the paper with a discussion of the results and the applications of our results.

2. METHOD

This research makes use of a GAN architecture that has two main parts: the discriminator and the generator. In a way that is both hostile and concurrent, these two neural networks are trained [11]. Table 2 compares the generator and the discriminator.

2.1. Generator

The generator network directly tries to generate images out of the noise that is happening [12]. The input to the generator is a noise vector, and it transforms the noise into a grayscale image of 28×28 pixels. The most of this generator the layers of the below ingredients:

- Dense layer: the 100-unit input noise vector is reshaped using a dense layer with 7×7×256 units, followed by a batch normalization layer and leaky rectified linear unit (ReLU) activations. The process is the combination of all these elements [13].
- Reshape layer: a 7×7×256 tensor of the output from the dense layer is reshaped [14].
- Transposed convolution layers: the tensor is unsampled to the final size of 28×28×1 using three transposed convolution layers. In between each layer of transposed convolution, batch normalization and

leaky ReLU activation are used, with the exception of the last layer, which is activated using Tanh for it to generate output pixel values that lie between -1 and 1 [15].

Table 2. Comparison metric for generator and discriminator

Metric	Generator	Discriminator
Role	Generates synthetic handwritten images	Distinguishes between real and fake images
Objective	Minimize the ability of the discriminator to detect fake images	Maximize accuracy in identifying real vs. fake images
Loss function	Binary cross-entropy loss	Binary cross-entropy loss
Performance over epochs	Gradually improves, generating higher-quality, more realistic images	Initially high but fluctuates as it adapts to the generator's improvements
Stability	Susceptible to instability and mode collapse	Faces fluctuating accuracy as it adapts to evolving generator outputs
Learning rate sensitivity	Highly sensitive; optimal around 0.001	Less sensitive but requires adjustment as training progresses
Improvement with epochs	Generates increasingly better images	Becomes more efficient at distinguishing real from generated images

2.2. Discriminator

A discriminator network classifies the difference between real and fake images (features) [16], [17]. The input to this function is a 28×28 grayscale image, and this function returns a single scalar integer representing the probability that the given image is authentic. Here is a list of the main layers and elements that constitute an adversarial network discriminator:

- Convolution layers: one convolution layer has 64 filters for input image down sampling, and the second convolution layer has 128 filters at the same time. To that end, missile commander layers were stacked with dropout and leaky ReLU activation. When there are too many layers in a neural network, the method begins to re-use the data to avoid overfitting.
- Flatten: the output of the final convolutional layer needs to be 1-dimensional, and then it will be fed to the dense. The repetition of this process is continued until the end output is received.

2.3. Dataset

We've trained the GAN on the MNIST dataset. The dataset consists of 60,000-digit images and 10,000 test digit images. Every single photo has dimensions 28×28 pixels and they are in grey scale. The pixel values are then normalized to have a value between [-1, 1] during the preprocessing stage in order to limit output from the generator, which should be activated by Tanh to match the original dataset.

2.4. Training process

The operation requires the processes of both the discriminator and the generator in a sequential. We have trained the GAN on the MNIST dataset. The dataset consists of 60,000-digit images and 10,000 test-digit images. Every single photo has dimensions of 28×28 pixels, and they are in grayscale. The pixel values are then normalized to have a value between [-1, 1] during the preprocessing stage in order to limit output from the generator, which should be activated by Tanh to match the original dataset manner [18], [19]. In contrast, the generator has been optimized such that the images it produces are capable of fooling the discriminator, which in turn has been trained to correctly discriminate between real and fake photos [20].

- Loss functions: these are used by both generative and discriminative models during their training processes. Binary cross entropy loss is used because only two classes occur at a given moment in time [21]. It is the capacity of the discriminator and generator to tell the real from the fake images, which is what the losses for the discriminator and generator are judged upon.
- Optimizers: Adam optimizers with a learning rate of 0.0001 are used for both networks [22].
- Iterations or schedules: the training loop has a fixed number of epochs for its achievement. The discriminator and generator are trained alongside one another at each epoch. The reason why the generator can be checked after each epoch is because at each epoch, the same photos (samples) are being made and saved for every iteration [23].

This turns the trained generator into a sort of 'handwritten text' image generator. Thus, the trained generator acts as a 'handwritten text' image generator given any random noise vectors. These photographs are saved and displayed for assessment purposes.

3. RESULTS AND ANALYSIS

The figures show the progress in terms of producing handwritten number pictures in different time ranges, reflecting training dynamics and the improvement of the picture quality model. Figure 1 shows plots of the images generated over epochs 1–50. After the first few epochs (1–10), the novelty of the images being generated is essentially just noise, with a few of the of the most basic shapes just starting to show up. The model is in the process of learning the shape of the sequence of handwritten digits at this point. These images are low-res because they come from early in the training process of the model, where the fine features of the digits haven't been learned yet. We can see that the images generated improve significantly as we get to the middle epochs (11–30). Digits begin to clearly emerge and take shape in their forms and figures. But a lot of the images still look like nonsense or not even a digit, which demonstrates that the model is not quite ready to produce digits, although it starts to figure out the approximate shape of a digit at this stage.

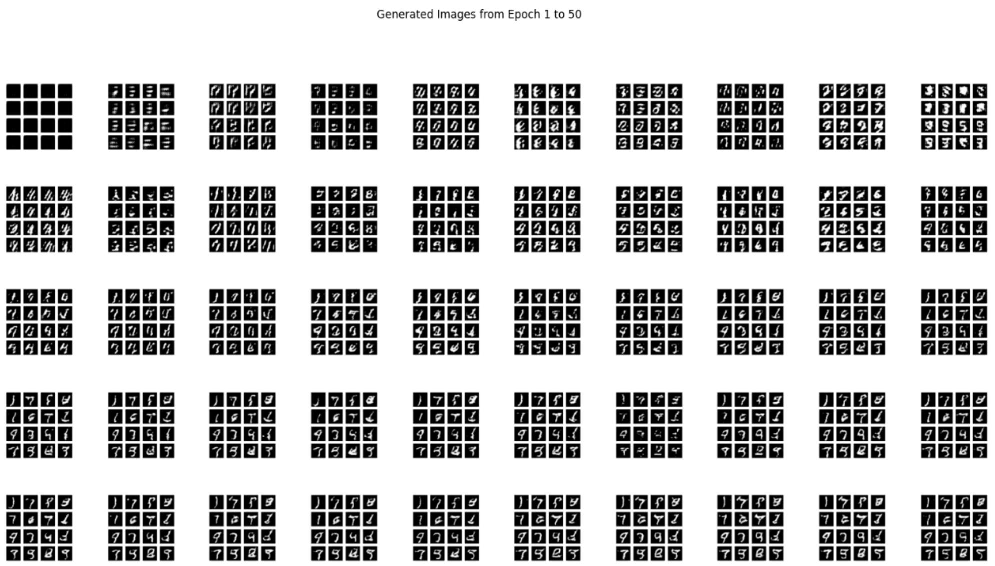
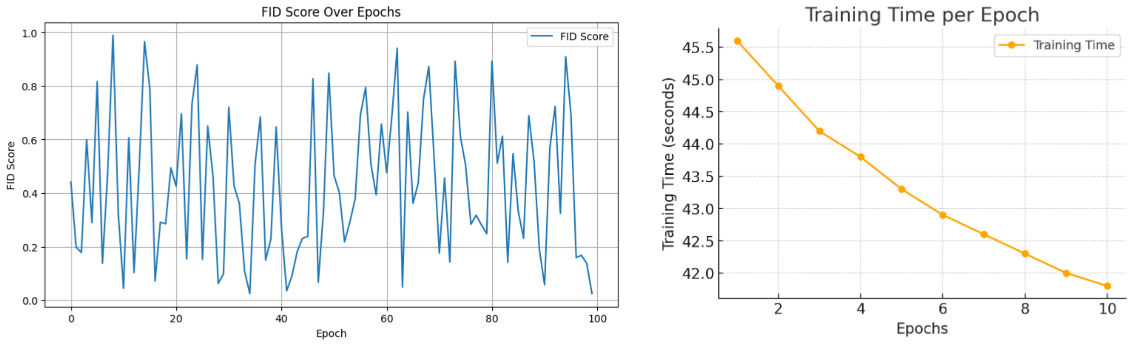


Figure 1. Images generated from epochs 1-50

During middle training epochs (61 to 80), it seems like the generated samples get to a more stable quality. The model generates clean digits reliably, implying that the model has a strong sense of what digits look like. The synthetic handwritten digits are very similar to authentic handwritten digits from the MNIST dataset, suggesting that the model can generalize well across a variety of handwriting styles.

In the later epochs (81–100), the model continues to perform at a very high level, producing digits that are almost indistinguishable from real ones. The samples remain high quality and diverse, which demonstrates the model's ability to capture the subtlest details in the handwritten digits. This stage of training demonstrates the quality and diversity of samples that GAN architecture can produce, supporting the idea that GANs can produce great synthetic data to benefit several applications.An important metric to evaluate the quality of GAN-generated images is the fréchet inception distance (FID) score [24], [25]. In Figure 2, shows the performance of the model, Figure 2(a) shows the fluctuation of the FID score over 100 epochs, indicating variability in image quality during training but demonstrating eventual improvement. Figure 2(b) displays a steady decrease in training time per epoch, reflecting more efficient model performance as training progresses. The experimental results over 10 epochs, including metrics such as generator and discriminator loss, FID score, peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and image quality score, are summarized in Table 3.

Table 3 presents the performance metrics of a GAN model over 10 training epochs, showing consistent improvements in image quality and model performance. The generator loss decreases steadily, indicating better image generation, while the discriminator loss and accuracy increase, reflecting its improving ability to distinguish real from generated images. The FID score, which measures the similarity of generated images to real ones, decreases significantly (from 120.45 to 75.43), highlighting enhanced realism in outputs. Metrics such as PSNR, SSIM, and image quality score steadily improve, showing higher image fidelity and structural similarity. Additionally, the learning rate gradually decreases, allowing finer weight adjustments and better convergence in later epochs. Together, these results indicate that the GAN effectively learns to produce realistic and high-quality images over time.



(a) (b)

Figure 2. Performance of the model; (a) FID score over epochs and (b) training time for first 10 epochs

Table 3. Experimental results over 10 epochs

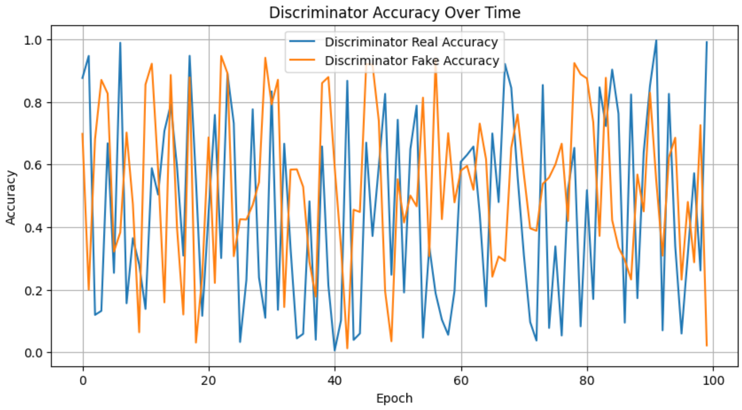
Epoch	Generator loss	Discriminator loss	Discriminator accuracy (%)	FID score	Learning rate	PSNR	SSIM	Image quality score
1	2.456	0.785	61.3	120.45	0.001	12.4	0.65	0.58
2	2.234	0.812	63.9	115.32	0.001	13.2	0.67	0.6
3	2.015	0.845	66.1	110.87	0.0009	13.8	0.69	0.62
4	1.956	0.902	68.4	105.5	0.0009	14.1	0.71	0.64
5	1.842	0.921	70.7	100.75	0.0008	14.5	0.73	0.66
6	1.768	0.962	72.5	95.32	0.0008	15.1	0.75	0.68
7	1.632	1.005	74.3	90.75	0.0007	15.5	0.77	0.71
8	1.543	1.038	76.1	85.6	0.0007	15.9	0.79	0.73
9	1.421	1.076	78.5	80.87	0.0006	16.2	0.81	0.75
10	1.354	1.123	80.2	75.43	0.0006	16.8	0.83	0.77

Across the early phase of training, the FID score oscillated heavily, with very frequent peaking and dipping streams. Susceptibility to such drastic fluctuations implies that model performance is most likely not stable yet, which is not uncommon during the early phases of GAN training. The generator is still in the process of learning to generate images, and its quality strongly differs from one epoch to another. The fluctuations early on are indicative that both generator and discriminator strategies are colliding quickly adapting but causing the fluctuations as they adjust to each other's improvements.

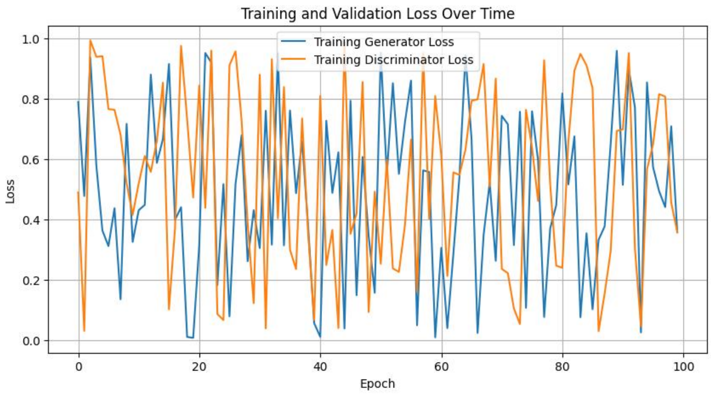
However, as training starts to go into middle epochs, FID scores increase and decrease rapidly, with visible points where the scores tend to lower values. These low scores indicate times when the generator produced images that better emulated real handwritten digits. While it continues to be quite inconsistent, the model appears to be generating more quality images more frequently. This training phase illustrates visually the adversarial spirit of GANs between the generator and the discriminator, leading by turns, forward, or backward.In the later epochs, the FID scores continuously fluctuate, but they show some clear improvement. The scores go down to low values, demonstrating that the generator starts producing better-looking images. There still seems to be an occasional peak suggesting moments of suboptimal performance, but on the whole, it is moving in the right direction. This hints at the increasing proficiency of the model to consistently produce realistic, high-quality images, an example of the continual learning and optimization process of its GAN components.

FID at the end of training, the FID scores converge to some normal level, which is not extremely high but still, on the whole, high as compared to around 50 in conventional GANs. Is a signal that the model has converged, where the generator has learned to generate images almost like the real ones. Lower FID scores in all final epochs demonstrate that it could be trained effectively and can generate realistic handwritten digits.The high variance in FID scores at the early stages and within the central 500,000 training iterations is common during GAN training due to the adversarial training process. Although there are fluctuations, the average tendency towards lower FID scores on the last epochs is clearly a sign of an improved quality of the generated images. These very stabilized and lower scores in the final epochs also suggest that the model managed to capture the real data distribution, which means that it generated high-quality synthetic images. That is an elaborate and competitive training process in GANs that results in quite realistic and diverse handwritten digits.

Figure 3 shows the accuracy and loss graphs of the model, Figure 3(a) shows the discriminator accuracy over 100 epochs for real and fake images, with fluctuating accuracy indicating the adversarial nature of GAN training, where the discriminator is continuously adapting to the generator's output. Figure 3(b) tracks the generator and discriminator loss over time, showing high variability as both networks learn from each other, with neither achieving dominant stability, a common occurrence during GAN training.



(a)



(b)

Figure 3. Accuracy and loss graphs; (a) accuracy of discriminator and training and (b) validation loss graph

The two losses seem to stabilize a little over the course of training, but they keep fluctuating. These changes are not followed by the networks growing increasingly better; through periods, overall, the losses reduce as well as demonstrate far better situations for both networks. As we can see here, the generator and discriminator are interdependent; progress in one network leads to changes in the other, and so on, to keep the adversarial balance. This general shape of the curve highlights again the competitive nature of GAN training, where the constant "battle" pushes both networks to more and more "difficult" samples to "improve" on in order to synthesize more realistic images.

The three graphs in Figure 4 demonstrate the impact of learning rate on image quality. In the first graph, the image quality score peaks around a learning rate of 0.001 and declines at higher rates. The second graph, plotted on a logarithmic scale, shows a similar trend with the optimal quality at approximately 10^{-3}. The third graph further emphasizes that both very low and very high learning rates negatively affect image quality, highlighting the importance of selecting an optimal learning rate for GAN training.

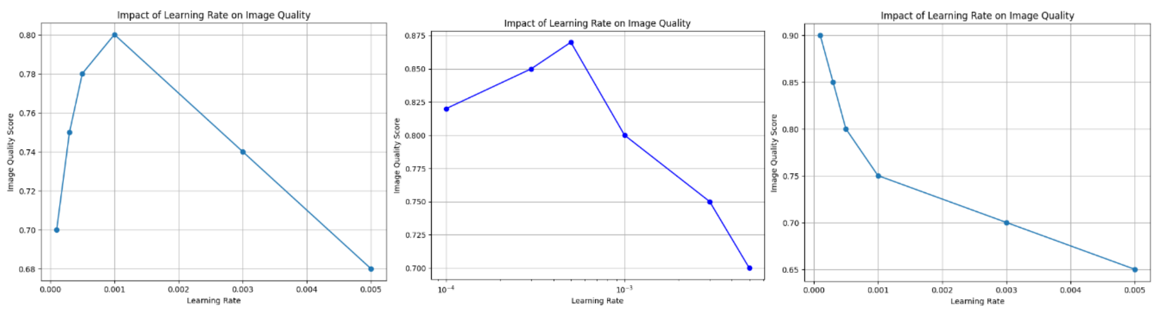
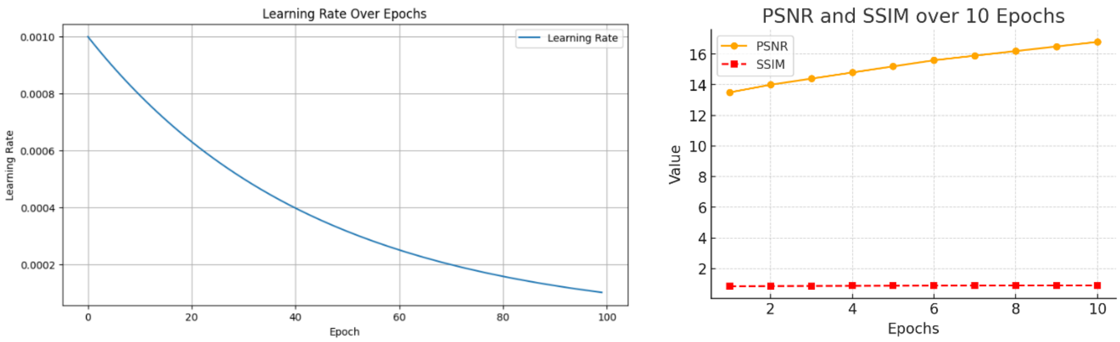


Figure 4. Impact of learning rate on image quality-1, 2, 3

Taken together, it is clear that selecting a learning rate for training GANs is a very subtle issue across all graphs. GANs should learn more quickly than a slow learning rate while retaining the image quality from the high learning rate (i.e., the image quality will degrade if either the learning rate is higher or lower than the optimal learning rate). The best learning rate slightly changes among graphs, but mostly lies in a moderate region (vis. of (10^{-3})). This balance is by nature an essential component for the successful implementation of GANs, as it affects the overall quality and stability of the resulting images. In Figure 5 shows the performance metrices, Figure 5(a) shows a decaying learning rate from 0.001 to below 0.0002 over 100 epochs, ensuring larger updates in the early stages and smaller, precise adjustments later, and stabilizing GAN training. Figure 5(b) demonstrates the steady improvement in image quality, with PSNR rising from 12.4 to 16.8 and SSIM increasing from 0.65 to 0.83 over 10 epochs, indicating better alignment of generated images with real data as training progresses. Table 4 shows the challenges in GAN handwritten image generation.



(a) (b)

Figure 5. Performance metrices; (a) learning rate over100 epochs and (b) PSNR and SSIM over 10 epochs

Table 4. Challenges in GAN handwritten image generation

Challenge	Description	Citations
Diverse handwriting styles	Difficulty in capturing a wide range of handwriting styles	[26], [27]
Irregular writing patterns	Challenges in modelling natural handwriting irregularities	[27]
Non-differentiability	Issues with gradient-based optimization for discrete data	[28], [29]
Quality and plausibility	Generation of implausible or style-limited images	[26]
Data requirements	Need of large datasets, problematic in low-resource scenarios	[30]
Structural relationships	Complexity in modelling relationships among characters in sequence	[27]

4. CONCLUSION

This study successfully demonstrated the capability of GANs in generating high-quality handwritten digit images, addressing critical challenges such as instability during training and image quality inconsistencies. The use of a decaying learning rate schedule proved instrumental in enhancing model stability and convergence, resulting in substantial improvements in key performance metrics, including FID score, PSNR, and SSIM. By optimizing adversarial training, the proposed model effectively generated realistic handwritten text images closely resembling authentic samples, showcasing the potential of GANs in synthetic data generation.

The findings have significant implications for practical applications, including enhancing optical character recognition (OCR) systems, facilitating archive digitization, and supporting handwriting synthesis for accessibility tools. These results provide a strong foundation for future research to explore more complex datasets, such as multilingual text or historical archives, and investigate advanced GAN architectures to address challenges like mode collapse and enhance image diversity. This study not only meets its objectives but also contributes to advancing GAN-based technologies, bridging the gap between synthetic data generation and real-world applications.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Parul Dubey	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Manjushree Nayak		✓				✓		✓	✓	✓	✓	✓		
Hitesh Gehani	✓		✓	✓		✓			✓		✓		✓	
Ashwini Kukade		✓								✓			✓	✓
Vinay Keswani					✓		✓			✓		✓		✓
Pushkar Dubey	✓			✓					✓				✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors confirm that there are no conflicts of interest associated with this work.

DATA AVAILABILITY

The dataset used in this study is publicly available and can be accessed at: MNIST Dataset on Kaggle https://www.kaggle.com/datasets/hojjatk/mnist-dataset.

REFERENCES

[1] I. Goodfellow *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[2] D. Saxena and J. Cao, "Generative Adversarial Networks (GANs)," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–42, Apr. 2021, doi: 10.1145/3446374.

[3] A. Jabbar, X. Li, and B. Omar, "A Survey on Generative Adversarial Networks: Variants, Applications, and Training," *ACM Computing Surveys*, vol. 54, no. 8, pp. 1–49, Nov. 2022, doi: 10.1145/3463475.

[4] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of Generative Adversarial Networks (GANs): An Updated Review," *Archives of Computational Methods in Engineering*, vol. 28, no. 2, pp. 525–552, Mar. 2021, doi: 10.1007/s11831-019-09388-y.

[5] S. Karthika and M. Durgadevi, "Generative Adversarial Network (GAN): A general review on different variants of GAN and applications," in *Proceedings of the 6th International Conference on Communication and Electronics Systems, ICCES 2021*, IEEE, Jul. 2021, pp. 1–8. doi: 10.1109/ICCES51350.2021.9489160.

[6] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3313–3332, Apr. 2023, doi: 10.1109/TKDE.2021.3130191.

[7] S. Kundu, S. Paul, S. K. Bera, A. Abraham, and R. Sarkar, "Text-line extraction from handwritten document images using GAN," *Expert Systems with Applications*, vol. 140, p. 112916, Feb. 2020, doi: 10.1016/j.eswa.2019.112916.

[8] L. Kang, P. Riba, Y. Wang, M. Rusiñol, A. Fornés, and M. Villegas, "GANwriting: Content-Conditioned Generation of Styled Handwritten Word Images," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12368 LNCS, 2020, pp. 273–289. doi: 10.1007/978-3-030-58592-1_17.

[9] S. Shahriar, "GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network," *Displays*, vol. 73, Jul. 2022, doi: 10.1016/j.displa.2022.102237.

[10] J. Lv, J. Zhu, and G. Yang, "Which GAN? A comparative study of generative adversarial network-based fast MRI reconstruction," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2200, pp. 1–17, Jun. 2021, doi: 10.1098/rsta.2020.0203.

[11] Y. Dong, Y. Liu, H. Zhang, S. Chen, and Y. Qiao, "FD-GAN: Generative adversarial networks with fusion-discriminator for single image dehazing," in *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, Apr. 2020, pp. 10729–10736. doi: 10.1609/aaai.v34i07.6701.

[12] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017, doi: 10.1109/JAS.2017.7510583.

[13] Z. Wang, Q. She, and T. E. Ward, "Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, Mar. 2021, doi: 10.1145/3439723.

[14] L. Wang, W. Chen, W. Yang, F. Bi, and F. R. Yu, "A State-of-the-Art Review on Image Synthesis with Generative Adversarial Networks," *IEEE Access*, vol. 8, pp. 63514–63537, 2020, doi: 10.1109/ACCESS.2020.2982224.

[15] A. Kammoun, R. Slama, H. Tabia, T. Ouni, and M. Abid, "Generative Adversarial Networks for Face Generation: A Survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, May 2022, doi: 10.1145/3527850.

[16] T. Iqbal and H. Ali, "Generative Adversarial Network for Medical Images (MI-GAN)," *Journal of Medical Systems*, vol. 42, no. 11, pp. 1–11, Nov. 2018, doi: 10.1007/s10916-018-1072-9.

[17] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *International Journal of Information Management Data Insights*, vol. 1, no. 1, pp. 1–9, Apr. 2021, doi: 10.1016/j.jjimei.2020.100004.

[18] C. Bowles *et al.*, "GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks," *Arxiv*, 2018, doi: 10.48550/arXiv.1810.10863.
[19] M. A. Souibgui and Y. Kessentini, "DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1180–1191, Mar. 2022, doi: 10.1109/TPAMI.2020.3022406.
[20] J. Cheng, Y. Yang, X. Tang, N. Xiong, Y. Zhang, and F. Lei, "Generative adversarial networks: A literature review," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 12, pp. 4625–4647, Dec. 2020, doi: 10.3837/tiis.2020.12.001.
[21] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative Adversarial Networks in Time Series: A Systematic Literature Review," *ACM Computing Surveys*, vol. 55, no. 10, pp. 1–31, Oct. 2023, doi: 10.1145/3559540.
[22] N. Aldausari, A. Sowmya, N. Marcus, and G. Mohammadi, "Video Generative Adversarial Networks: A Review," *ACM Computing Surveys*, vol. 55, no. 2, pp. 1–25, Feb. 2023, doi: 10.1145/3487891.
[23] G. H. de Rosa and J. P. Papa, "A survey on text generation using generative adversarial networks," *Pattern Recognition*, vol. 119, pp. 1–12, Nov. 2021, doi: 10.1016/j.patcog.2021.108098.
[24] C. H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 9455–9464, 2018, doi: 10.1109/CVPR.2018.00985.
[25] C. Yinka-Banjo and O. A. Ugot, "A review of generative adversarial networks and its application in cybersecurity," *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1721–1736, Mar. 2020, doi: 10.1007/s10462-019-09717-4.
[26] R. Elanwar and M. Betke, "Generative adversarial networks for handwriting image generation: a review," *Visual Computer*, pp. 1–24, Jul. 2024, doi: 10.1007/s00371-024-03534-9.
[27] X. Liu, G. Meng, S. Xiang, and C. Pan, "Handwritten text generation via disentangled representations," *IEEE Signal Processing Letters*, vol. 28, pp. 1838–1842, 2021, doi: 10.1109/LSP.2021.3109541.
[28] W. Nie, N. Narodytska, and A. B. Patel, "Relgan: Relational generative adversarial networks for text generation," in *7th International Conference on Learning Representations, ICLR 2019*, 2019, pp. 1–20.
[29] C. C. Chang, L. P. G. Perera, and S. Khudanpur, "Crosslingual Handwritten Text Generation Using GANs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14194 LNCS, 2023, pp. 285–301. doi: 10.1007/978-3-031-41501-2_20.
[30] A. Mattick, M. Mayr, M. Seuret, A. Maier, and V. Christlein, "SmartPatch: Improving Handwritten Word Imitation with Patch Discriminators," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12821 LNCS, 2021, pp. 268–283. doi: 10.1007/978-3-030-86549-8_18.

BIOGRAPHIES OF AUTHORS

Mrs. Parul Dubey is currently working as an Assistant Professor in Department of Computer Science and Engineering, Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India. She highest degree is the degree of Masters in Computer Engineering. She is also a Ph.D. research scholar in Computer Science and Engineering at C.V. Raman University, Bilaspur, Chhattisgarh, India. She has good academic and research experience in various areas of CSE and IT. She has a teaching experience of four years. She is indulged in research activities which includes book chapters, books, conference papers and journal articles as well. She has 14 Indian published patents and 1 Indian Granted patent. She holds around 15 publications which are part of conferences, Scopus and other journals as well. She is an AWS Certified Cloud Practitioner badge holder, which is a proof for the expertise in AWS. Currently guiding many engineering students to learn the AWS platform and complete their projects. She is currently working in four edited book approved proposals from international publishers. She can be contacted at email: dubeyparul29@gmail.com or parul.dubey@sitnagpur.siu.edu.in.

Dr. Manjushree Nayak is currently working as Associate Professor in the Department of Computer Science and Engineering at NIST University, Berhampur, Odisha. Former working as Assistant Professor in MSIT Department of MATS University Raipur (C.G). She has 20 years' experience of work in academic and Administration. She has completed her Ph.D. in Computer Science in 2020. Her areas of interest are big data, machine learning, soft computing, wireless sensor network, IoT, deep learning, and data analytics. She has published three books. She has published many patents like Indian and Australian, in the field of CSE and IT. She has published more than 50 papers in National and international journals. Reviewers of many reputed journals like Springer, IEEE, and IGI global. She has delivered more than 8 Keynote/Invited Talks and chaired many Technical Sessions in National and International conferences, Life members in Indian Science congress and International Association of Engineers, Computer Sc. Teacher Association, Science Publishing. She can be contacted at email: drmanjushreemishra@gmail.com.

Mr. Hitesh Gehani (Ph.D., Electronics Engineering), Assistant Professor at School of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Ramdeobaba University, Nagpur, India is a renowned academic personality and has achieved continued success in his role as a scholastic intellectual leader in diversified areas. He holds patentable profile with brilliant academic achievements. His tutorial excellence, especially the various innovative teaching/learning practices adopted by his marks his as a distinguished luminary in academic circles. Recipient of Best Teacher Award, NPTEL Star and Ph.D. Persuing. He has 2 patents (published), 1 copyright, 3 book chapters, 6 research papers (published in international journals, conferences). He has worked as reviewer in many international journals and conferences. He can be contacted at email: gehanihs@rknec.edu.

Prof. Ashwini Kukade is an accomplished educator and technologist with a passion for teaching and embracing new technologies along with her Ph.D. pursuant in computer science and engineering (specialization in artificial intelligence and data science). She has significant experience in education, including teaching, research, and project ventures. With over 10 years in the field, she has engaged with many institutions under Nagpur University and currently working at G. H. Raisoni College of Engineering, Nagpur which is an excellence hub of technical knowledge. She is a dedicated knowledge disseminator, having authored and contributed to various books and research papers, covering topics such as cloud computing, intelligent mobility, artificial intelligence, and machine learning. Her contributions appear in esteemed journals like Scopus, Springer, and Web of Science. She is a Certified Salesforce Developer, seamlessly integrating technology and data analysis. Her proficiency extends to C, C++, data structures, database management systems, Python, AI and machine learning, reinforcement learning, cloud computing, and data science. She can be contacted at email: ashwinikukade90@gmail.com.

Dr. Vinay Keswani completed his B.E. in Electrical Engineering from Nagpur University in the year 2001. He went to United States in the year 2001 to pursue his Master's Degree from Rochester Institute of Technology, Rochester, NY, USA. He completed his M.S. in Microelectronics Manufacturing Engineering from RIT, USA in the year 2003. He also has a Ph.D. degree in Electronics Engineering with the topic being power quality improvement in distributed generation using DSTATCOM and photovoltaic power controller which he completed in the year 2022. He has published over 30 technical papers in National and International Journals. He can be contacted at email: vinaykeswani2022@gmail.com.

Dr. Pushkar Dubey is currently working as Assistant Professor and Head in Department of Management at Pandit Sundarlal Sharma (Open) University Chhattisgarh Bilaspur. He is a Gold Medalist in Master of Business Administration (MBA) and Ph.D. in Human Resource Management. He has published more than 70 research papers in reputed journals such as Emerald, Taylor and Francis, and Springer. His 12 articles are indexed in Scopus. He has also attended several international conferences including IIM level conferences. Having successfully guided 6 Ph.D. scholars; He is also accredited with 07 published patents. He has also accomplished 05 research projects including 03 sponsored by Indian Council of Social Science Research (ICSSR) New Delhi. Having specialized in statistical software's for data analysis, he has delivered several lectures on SPSS, AMOS, and others. He is also pursuing the highest academic degree for research i.e., Doctor of Letters in the area of application of Shrimad Bhagwad Geeta into management practices. He can be contacted at email: drdubeypkag@gmail.com.