Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 3, June 2026, pp. 2495~2504
ISSN: 2302-9285, DOI: 10.11591/eei.v15i3.10358

Generating synthetic strike-through handwritten text using generative adversarial networks

Dheemanth Urs Raje, Chethan Hasigala Krishanappa
Department of Computer Science and Engineering, Maharaja Institute of Technology affiliated to Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Mar 26, 2025
Revised Mar 31, 2026
Accepted Apr 19, 2026

Keywords:

Document processing
Generative adversarial networks
Handwritten text recognition
Strike through text
Synthetic dataset

ABSTRACT

The evaluation of handwritten text documents is a significant research field in text analysis. Deep learning classifiers' effectiveness tends to decline when faced with variations in text style and the presence of strike-out text components. These strike-outs can occur at the character, word, paragraph, or page level. When these documents undergo optical character recognition (OCR) processing, they often yield inaccurate results. Data unavailability, imbalance poses a strong risk in this area. Hence usage of synthetic datasets for conducting research can solve the issue to an extent. Despite using traditional data augmentation techniques like rotation, position shifting, zooming, and shearing, the issue of imbalanced class distribution remained unchanged. In order to tackle this challenge this paper demonstrates the creation of a synthetic dataset comprising of strikethrough text using modified version of generative adversarial networks (GANs) that has an auxiliary network for text recognition. IAM and RIMES dataset are used as base for the GAN to generate synthetic images of handwritten text. A line segmentation technique is also implemented to create strike outs on random words selected from the synthetic image set thereby increasing the number of strikes. The work believes that the simulated dataset will significantly improve the quality of handwriting text recognition models.

Corresponding Author:

Dheemanth Urs Raje
Department of Computer Science and Engineering
Maharaja Institute of Technology affiliated to Visvesvaraya Technological University
Belagavi, India
Email: dheemu.urs@gmail.com

1. INTRODUCTION

Over the years, the area of automatic handwritten text recognition (HTR) has gained popularity due to digitization and the advancement of artificial intelligence. HTR finds application in health care, finance, and education. A doctor's prescription on paper is mostly handwritten. Many financial statements are hand-written, and similarly, answer sheets are handwritten by students. Handwritten documents contain multiple types of noise in it, one such noise is struck out words in the document. A lot of knowledge is struck in hand written text which has to be extracted and used for taking future decisions. Identifying the struck-out words in the handwritten documents has applications in the field of forensic science and human behavior analysis. Optical character recognition (OCR) is used for recognising printed text where as HTR is recognizing text in handwritten documents. The field of OCR has stabilised, but HTR is still lagging due to unique handwriting styles of each individuals. HTR has to deal with complex and novel challenges. One challenge of HTR is the presence of struck-out words in the text. The accuracy of a HTR comes drastically down if it comes across struck out words in the hand written text. In order to make the HTR effective it has to be trained on large

volumes of hand written text documents that have struck out words. The field is paralyzed by the unavailability of enough data that is annotated data. Hence, there is a need to create a synthetic dataset of handwritten text with struck-out words.

A group of scientists [1] performed a comprehensive survey on online and offline handwriting recognition techniques. Their study focused on pre-processing of images, word and character recognition, signature identification, and writer recognition. Poznanski and Wolf [2] estimated the n-grams of images using deep learning techniques. Later they compared these n-grams with already available words. A group of researchers [3] made advances in the work by using a pyramidal histogram of characters for detecting words in handwritten documents. A group of investigators [4] merged encoder-decoder with convolutional neural network (CNN) for identifying words in handwritten documents. The encoder-decoder pair was later named as sequence to sequence. Dutta *et al.* [5] performed handwriting identification using a hybrid model comprising of CNN as well as recurrent neural networks (RNN).

Computer-aided handwriting generation is a relatively new area with a limited amount of related work available. Nevertheless, this paper in this section tries to survey the available literature in this area. Fogel *et al.* [6] proposed a generative adversarial network (GAN)-based handwriting generation technique that can generate images of words with random lengths. They used semi-supervised learning techniques for generating synthetic images of handwritten text. Alonso *et al.* [7] proposed an adversarial model for synthesizing pictures of handcrafted word images using recurrent layers. The image dataset created by this work is free from all imbalances and contains French and Arabic words. Zdenek and Nakayama [8] proposed a model called JokerGAN that requires less memory and uses GANs with several categories of conditional group normalization. This model is capable of training on languages that have a lengthy character set. HiGAN [9] was proposed by a few researchers for generating images of human handwriting. This GAN is capable of simulating handwritten word images in different calligraphic styles. The text can be of flexible lengths. Transformer-based handwritten text image synthesis was introduced in [10], which is capable of generating images using local and global patterns. This model, when compared with other existing techniques, showed superior performance and quality. Sasipriyaa *et al.* [11] used scrabble GAN and CNNs to generate handwritten text images of the Tamil Language. This dataset was free from imbalances and acted as a good base for training and testing other handwritten recognition models.

Data augmentation is also one technique for generating synthetic and large image datasets. GANs are consistently being used to do this activity. In this process, the true training image set is merged with the synthetic dataset thereby increasing the volume of the dataset. A low-shot training process was developed for data augmentation and preventing loss in a foolproof manner [12]. Bhunia *et al.* [13] proposed uses GAN to bind the old images with new images using a module of parameters. The inability to generate images outside a predefined lexicon is a serious drawback of this approach. In another attempt Krishnan *et al.* [14] to create a synthetic dataset through augmentation, researchers used deep learning methods for word identification. A group of researchers identified the basics, working, and challenges of GANs [15]. Urs and Chethan [16] conducted a comprehensive survey on available datasets, which clearly states the availability of a limited number of struck-out words, which leads to a bias towards a specific class of struck out words.

The IAM dataset [17] comprises 9,862 text lines, which are further divided into 6,161 lines for training, 900 lines for validation-1, 940 lines for validation-2, and 1,861 lines for testing. It encompasses 79 different symbols, including lowercase and uppercase letters, digits, punctuation marks, and the symbol "#" representing struck-out words. The RIMES dataset [18] offers French handwritten texts and includes a total of 129,414 images, 6,780 words, and 86 characters. The dataset is known as Reconnaissance et Indexation de données Manuscrites et de fac similÉS. The CVL database [19] serves as a public resource for writer retrieval, writer identification, and word spotting tasks. It consists of seven distinct handwritten texts, one in German and six in English. A total of 310 writers participated in the dataset, with 27 writers contributing seven texts and 283 writers contributing five texts.

The available datasets suffer from a significant class imbalance issue, as there are very few instances of struck-out words. This skewed distribution leads to biased predictions by classifiers, favoring the majority class. The length of the majority class gradient holds more influence than that of the minority class, resulting in the domination of the net gradient. Consequently, the convergence of the minority class is slower. Additionally, the limited number of struck-out words in the database is inadequate for effectively training the model. A large, balanced training dataset is required to solve these problems. Hence, this work uses a data augmentation technique to increase the size of the dataset as well as make it balanced. The new synthetic dataset is created by taking IAM and RIMES as a base. Table 1 suggests, the existing augmentation techniques prove to be ineffective in the case of classifying strike-out text. This paper proposes a new augmentation method.

In this research, we aim to bridge this gap by generating realistic synthesized handwritten text. This approach reduces the reliance on annotated data and enhances the variety of training data in terms of writing

styles and vocabulary. We employ a novel architecture that can generate words of arbitrary length and even complete sentences. The proposed architecture does not require character-level annotations and learns character embeddings autonomously. Words are then picked randomly and applied to various forms of strikes using line segmentation techniques. This way, a corpus of handwritten text images with struck-out words are synthesized. The rest of the paper is organized as follows: section 2, the methodology for synthesizing a strike through handwritten text is discussed. In section 3, results are portrayed and analysed for further processing. Finally, in section 4, conclusions and future works are drawn out.

Table 1. Strike-classification performance with various techniques along with traditional data augmentation

Methods	Non-strike-out	Strike-out
Traditional data augmentation	0.97	0.42
Ensemble technique	0.93	0.7
Cost-sensitive learning	0.91	0.48
Focal loss	0.84	0.17
K-stratified fold	0.92	0.43

2. METHOD

This study utilizes the GAN methodology to create a synthetic dataset of handwritten images, specifically featuring words that are struck out. The generator is designed to produce synthetic images resembling real handwritten text, while the discriminator evaluates these images to distinguish between real and synthetic data. Through iterative training, the generator improves its ability to create high-quality, realistic images, and the discriminator becomes better at identifying artificial data. The pipeline for synthetic strike-through handwriting dataset generation is depicted in Figure 1. After the generation of synthetic dataset, the quality of the images has to be validated using metrics like geometric score (GS) and Fréchet inception distance (FID). The rest of the section describes the various stages of the pipeline.

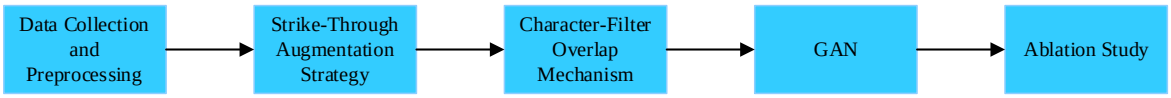

Figure 1. Pipeline for synthetic strike-through handwriting dataset generation

2.1. Data collection and preprocessing

The RSICB256 dataset is the foundation of this study. This dataset is a collection of diverse handwritten text samples. As part of preprocessing the data, the study removed illegible or excessively noisy entries. Also, all the images were resized to 256×256 pixels, normalized to the range [0,1], and segmented into text lines to ensure consistency across experiments.

2.2. Strike-through augmentation strategy

In order to simulate realistic strike-through handwriting, the study developed a stroke augmentation pipeline. The different types of strike-through styles are generated programmatically [20] and superimposed on the handwritten text lines. Table 2 illustrates the augmentation pipeline developed for the study.

Table 2. Augmentation pipeline

Type of strokes	Strategy
Simple strokes	Single straight, left-slanted, right-slanted, and crossed lines.
Multiple strokes	Parallel, crossed, and slanted variations.
Box strokes	Enclosed strike-throughs in straight, slanted, or crossed styles.
Irregular strokes	Zig-zag, wavy, hybrid, and blackout patterns.

2.3. Strike-through augmentation strategy

In order to improve realism in strike-through placement a character filter overlap mechanism is implemented in the study. The overlapping filters ensure partial coverage across characters, mimicking natural handwriting variations. The mechanism consists of three stages namely overlapping extent, selection, and implementation. Experiments were conducted with 10%, 25%, and 50% of the character width. The script characteristics were used to choose the overlap size. Finally, a sliding window filter was applied across segmented characters, dynamically adjusting strike-through positioning.

2.4. Strike-through augmentation strategy

This work employed a conditional GAN (cGAN) [21] architecture to synthesize realistic struck-through handwritten text images: the generator is responsible for superimposing strike-through strokes on clean handwritten text. On the other hand, the discriminator is trained to differentiate between real (human-written) and synthetic (GAN-generated) strike-through text. For maintaining realism and structural accuracy of the images this work used adversarial and L1 reconstruction loss. The hyper-parameters selected for training are batch size of 32, learning rate of 0.0002, and Adam optimizer.

The generator [22] developed is novel in terms of its technical elements which will be described below. Firstly, the number of characters in the input word is identified. An equivalent number of character filters are considered. These filters are merged. A noise vector is calculated and multiplied with the concatenated filters. The image generated by this process is given as input to the generator. The generated image is given as input to the discriminator for determining whether the image was original or synthesized. The generator used is class-conditioned. Every character is generated individually, which sets the methodology apart from other handwriting techniques available in the literature. The generator acts as an addition of identical character conditional generators. This overlap permits neighbouring characters to mingle for creating a smooth change. Here the character filters learn from dependencies from adjacent character filters. The generator learns to create variations of the same character by using these dependencies. Five-character filters are concatenated for the word "SALUT". Let 'n' be the noise vector, and it is multiplied to the concatenated image resulting from five filters. If $\{f_s, f_a, f_l, f_u, f_t\}$ represent the character filters, then n x $\{f_s, f_a, f_l, f_u, f_t\}$ is input to the generator. Figure 2 shows the overlapping of character filters.

The role of the discriminator [23] is to judge whether the generated images are true or fake. This work uses a CNN for designing the discriminator. The CNN is formed from a combination of real/fake classifiers as well as a pooling layer. Figure 3 shows the GAN architecture overview for generating a word like "SALUT", while Figure 4 presents sample handwritten images generated by the proposed GAN architecture.

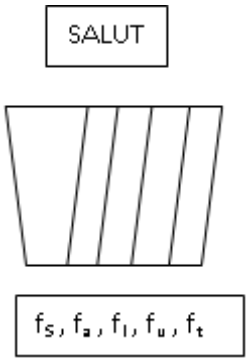

Figure 2. Overlapping of character filters

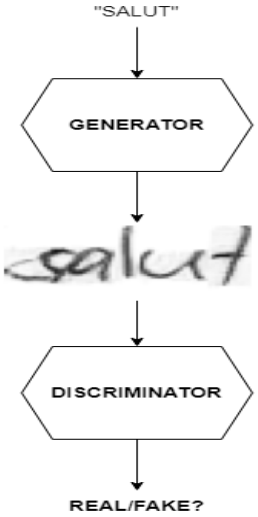

Figure 3. GAN architecture overview for the case of generating the word "SALUT"

Figure 4. Sample handwritten images generated by the proposed GAN architecture

2.5. Ablation study

To validate the contribution of the character-filter overlap, we conducted ablation experiments with four settings namely no overlap, 10% overlap, 25% overlap, and 50% overlap. In the no overlap strike-through strokes were placed independently without overlapping filters. A minimal overlap introducing slight character occlusion was maintained in 10% overlap setting. A moderate overlap was maintained in 25% overlap category. This setting balanced the natural appearance and readability. The 50% overlap setting lead to aggressive occlusion.

The effect of character-filter overlap on synthetic image quality is illustrated in Table 3. SSIM stands for structural similarity index measure. It is a metric for analyzing the similarity of two images in terms of luminance, contrast and structure. FID [24] is a metric used to evaluate the quality of generated images by comparing the statistics of the generated images and the real images. It measures the similarity between the feature representations extracted from the generated and real images using a pre-trained Inception model. From the Table 3 it is evident that 25% overlap attains a maximum SSIM of 0.93 and a minimum FID of 32.1 when compared with other categories. Hence the study selected 25% overlap as it maintains optimal balance between readability and realism.

Table 3. Effect of character-filter overlap on synthetic image quality

Overlap level (%)	Readability (SSIM ↑)	Realism (FID ↓)	GAN stability	Observations
No overlap	0.86	45.2	Stable	Strokes appear artificial and lack natural alignment
10 overlap	0.9	38.7	Stable	Good for disconnected scripts; minor occlusion
25 overlap	0.93	32.1	Stable	Optimal balance; realistic strokes without loss of clarity
50 overlap	0.78	50.6	Unstable	Excessive occlusion; GAN convergence degraded

3. RESULTS AND DISCUSSION

In this section, the effectiveness of the proposed GAN-based architecture for generating offline handwritten text images is discussed in detail. The following subsection contains a detailed explanation of the implementation process and evaluation metrics. Additionally, the results are analyzed to demonstrate the quality and realism of the generated handwritten text images. The GAN training used a Pix2Pix-style cGAN. The cGAN comprised of a U-Net generator and PatchGAN discriminator. It was optimized with adversarial and L1 losses for 120 epochs. A batch size of 32 and learning rate of 0.0002 were also selected as settings.

3.1. Implementation details

The image processing system is specifically optimized to handle images with a constant height of 32 pixels. Moreover, the generator network (G) is structured to have a receptive field width of 16 pixels. In this configuration, the generator architecture is composed of a filter bank (F) representing the lowercase English alphabet. Each filter in the filter bank has dimensions of 32×8192. To generate a word with a length of n characters, the generator selects and concatenates n filters from the filter bank. These filters are then multiplied with a 32-dimensional noise vector (z1), resulting in an n×8192 matrix. This matrix is subsequently reshaped into a 512×4×tensor, where each character occupies a spatial size of 4×4. The tensor is then processed through three residual blocks that gradually increase the spatial resolution, introduce the

desired receptive field overlap, and eventually yield the final image size of 32×16n. To modulate the residual blocks, conditional instance normalization layers are utilized, incorporating three additional 32-dimensional noise vectors (z2, z3, and z4). Finally, a convolutional layer with a hyperbolic tangent (tanh) activation function is employed to produce the output image. Figure 5 shows the different styles of a single word ground truth "*pneumonoultramicroscopicsilicovolcanokoniosis*".

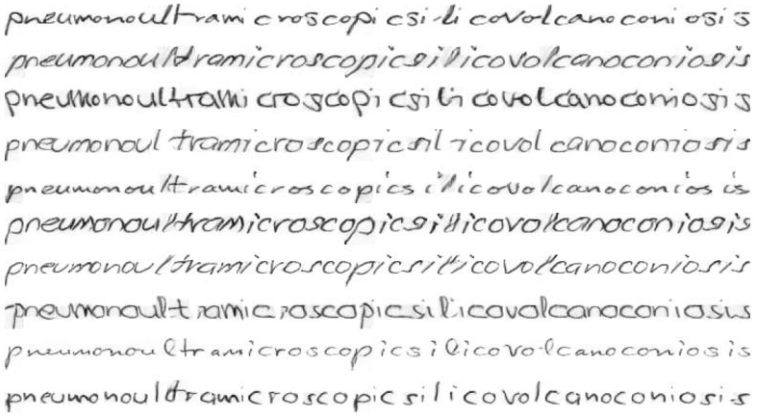

Figure 5. Different styles of a single word ground truth "*pneumonoultramicroscopicsilicovolcanokoniosis*"

The discriminator network (D) takes inspiration from BigGAN and consists of four residual blocks followed by a linear layer with a single output. To accommodate varying width image generation, the discriminator operates on horizontally overlapping image patches. The final prediction is obtained by averaging the predictions of these patches, which are then used in a GAN hinge-loss function.

During the training of the GAN, all images underwent resizing to a consistent height of 32 pixels while maintaining the aspect ratio of the original image. In the supervised scenario where labels were employed, an additional horizontal scaling was implemented on the images. This scaling ensured that each character in the real samples had a width of approximately 16 pixels, matching the width of the synthetic characters. The objective of this adjustment was to intensify the challenge for the discriminator by increasing the similarity between the real and synthesized samples. Experiments were conducted on an NVIDIA V100 GPU (16 GB RAM) using PyTorch. Datasets included IAM and RIMES, their augmented variants (IAM*, RIMES*), and the combined synthetic dataset. Splits were writer-disjoint to avoid stylistic overlap. Pre-processing involved grayscale conversion, binarization for line segmentation, and normalization to [0,1]. The augmentation of the strike-through images was performed by randomly selecting thickness, opacity, angle, and jitter.

3.2. Evaluation metrics

There are two popular methods using standard metrics for GAN performance evaluation, namely FID and GS [25]. The FID [26] is a metric used to evaluate the quality of generated images by comparing the statistics of the generated images and the real images. It measures the similarity between the feature representations extracted from the generated and real images using a pre-trained Inception model. The GS, on the other hand, is a metric that assesses the performance of the generator in a GAN. It measures how well the generator can deceive the discriminator and produce realistic and high-quality images. Higher GS values indicate better performance of the generator. The lower the FID value, the better the quality and diversity of the generated images.

$$\text{FID} = ||\mu\_\text{real} - \mu\_\text{fake}||\text{^}2 + \text{Tr}\ (\Sigma\_\text{real} + \Sigma\_\text{fake} - 2(\Sigma\_\text{real}\Sigma\_\text{fake})\text{^}0.5) \quad (1)$$

$$GS = log(D(G(z))) \quad (2)$$

Where, μ_real and μ_fake are the mean feature representations of real and generated images, respectively. Σ_real and Σ_fake are the covariance matrices of the feature representations of real and generated images, respectively. Tr () represents the trace of a matrix. ||...||^2 denotes the Euclidean distance squared. G(z) represents the generated image from a random noise vector z. D(.) is the discriminator network that assigns a

probability score to the generated image. log () denotes the logarithm. Table 4 depicts the FID and GS values obtained for the images generated by the proposed GAN architecture. The proposed GAN paradigm achieves an FID of 23.64 and GS of 6.5×10^{-4}. If GS is low, it indicates generated data aligns better with the real data manifold and if it is high, it indicates data mismatch.

Table 4. FID and GS values for the proposed GAN paradigms

Metric	Proposed GAN	IAM*Baseline	RIMES*Baseline
FID↓	23.64	32.8	34.1
GS ↓	6.5×10^{-4}	6.5×10^{-4}	6.5×10^{-4}

From the IAM training and validation sets, 30% and 25% of samples were used to generate struck-out texts, respectively. Similarly, from the RIMES training and validation sets, 30% and 25% of samples were used to create struck-out texts, respectively. The statistics of the original and modified datasets are displayed in Table 5.

Table 5. Statistics of IAM, IAM*, RIMES, RIMES*, and the synthetic dataset

	IAM	IAM*	RIMES	RIMES*	Synthetic dataset (IAM*+RIMES*)
No. of lines in training	8862	10762	1,78,122	231558	242320
No. of lines in validation	1000	1280	25,446	29976	31256
No. of lines in test	3491	5352	50,892	56784	62136
No. of struck-out words in training	50	1950	68,264	88024	89974
No. of struck-out words in test	0	480	17,066	24022	24022
No. of non-struck-out words in training	53757	67797	10,57,492	12,53,484	13,21,281
No. of non-struck-out words in test	17560	17080	2,64,374	3,45,247	3,62,327

In Figure 6 the statistics related to IAM, IAM*, RIMES, RIMES*, and the synthetic dataset are illustrated. The blue bars represent the training lines, orange bars represent the test lines, green bars represent the struck-out words in training and red bars represent the non-struck-out words in training. Figure 7 illustrates sample struck-out words from the synthetic dataset, such as the word "formaliser," showcasing the realism and diversity of the generated data.

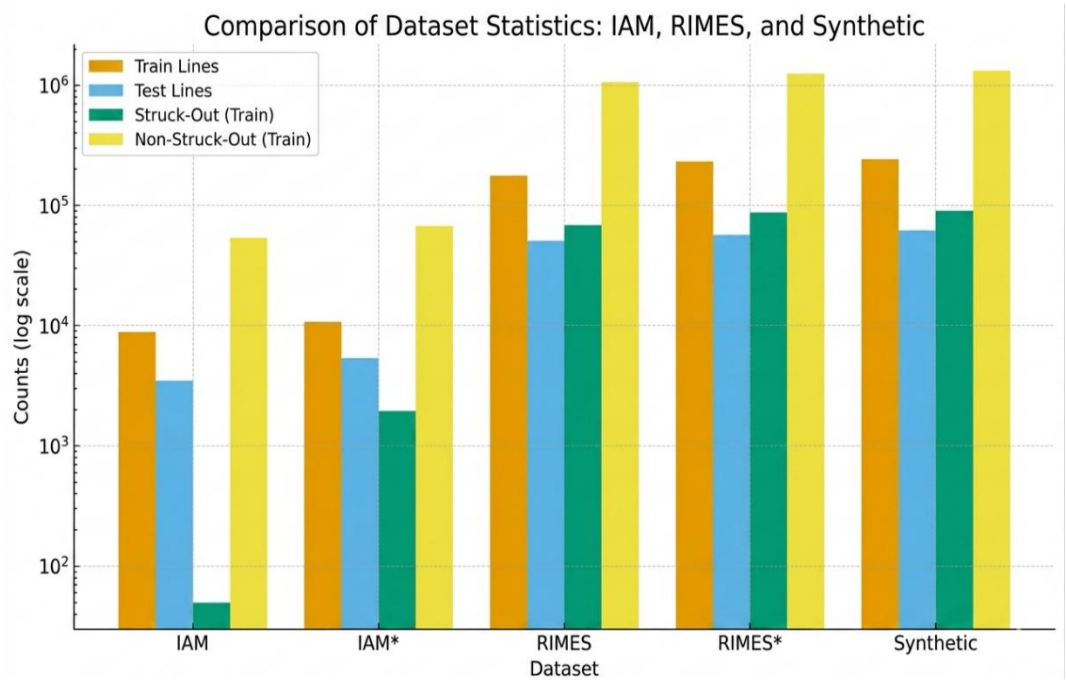

Figure 6. Comparison of dataset statistics

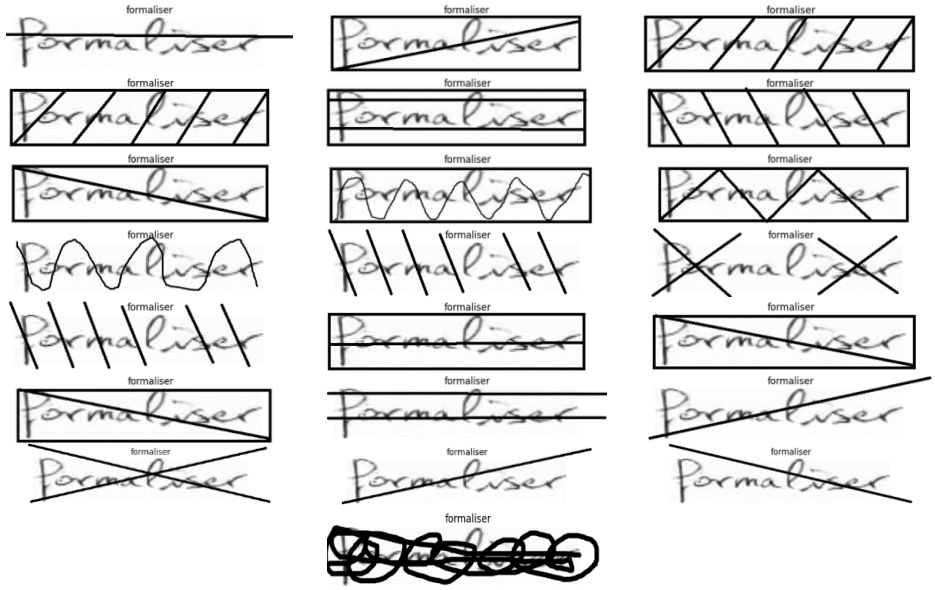

Figure 7. Sample struck out words from the synthetic dataset ground truth-"formaliser"

4. CONCLUSION

This paper introduced a novel GAN-based architecture for generating offline handwritten text images based on the concept that writing characters is a localized process. The proposed approach involves constructing each word by combining individual character images. The generated images exhibit versatility in stroke widths and overall style. Moreover, the overlap between the receptive fields of different characters within the text allows for the generation of both cursive and non-cursive handwriting. The paper also used a line segmentation method to create strike-out words in various patterns. Two datasets, IAM, and RIMES, were modified to IAM* and RIMES* by increasing the number of strike-out words in the datasets using GANS. Later the two IAM* and RIMES* were merged to create a synthetic dataset, which is balanced in all aspects and can be used with any HTR system. The proposed GAN achieved an FID of 23.64 and a GS of 6.5×10^{-4}, demonstrating that the synthetic strike-through samples are both visually realistic and geometrically well-aligned with the real data distribution. Compared to IAM*/RIMES* baselines, the proposed approach significantly improves both perceptual realism and distributional alignment. Our study believes that the extensive variability in the words and styles generated can enhance the performance of a given HTR system by augmenting the training dataset. These findings demonstrate that synthetic augmentation with controlled strike-throughs can improve the robustness of handwriting recognition systems, particularly for noisy real-world documents. While vocabulary coverage and writer style variability remain somewhat limited compared to real datasets, the pipeline offers a scalable way to generate realistic corrections and can be extended to multilingual scripts with different GAN architectures in future work.

ACKNOWLEDGMENTS

Primarily, we extend our heartfelt thanks to my guide for his guidance, valuable insights, and encouragement throughout the research process.

FUNDING INFORMATION

No funding is raised for this research.

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
Dheemanth Urs Raje	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
Chethan Hasigala Krishanappa	✓	✓			✓	✓		✓	✓	✓	✓	✓		

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

Authors state no conflict of interest.

DATA AVAILABILITY

Derived data supporting the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000, doi: 10.1109/34.824821.

[2] A. Poznanski and L. Wolf, "CNN-N-Gram for HandwritingWord Recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2305–2314, doi: 10.1109/CVPR.2016.253.

[3] S. Sudholt and G. A. Fink, "PHOCNet: A deep convolutional neural network for word spotting in handwritten documents," in *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, Oct. 2016, vol. 0, pp. 277–282, doi: 10.1109/ICFHR.2016.0060.

[4] J. Sueiras, V. Ruiz, A. Sanchez, and J. F. Velez, "Offline continuous handwriting recognition using sequence to sequence neural networks," *Neurocomputing*, vol. 289, pp. 119–128, May 2018, doi: 10.1016/j.neucom.2018.02.008.

[5] K. Dutta, P. Krishnan, M. Mathew, and C. V. Jawahar, "Improving CNN-RNN hybrid networks for handwriting recognition," in *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, Aug. 2018, pp. 80–85, doi: 10.1109/ICFHR-2018.2018.00023.

[6] S. Fogel, H. Averbuch-Elor, S. Cohen, S. Mazor, and R. Litman, "Scrabblegan: Semi-supervised varying length handwritten text generation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 4323–4332, doi: 10.1109/CVPR42600.2020.00438.

[7] E. Alonso, B. Moysset, and R. Messina, "Adversarial generation of handwritten text images conditioned on sequences," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, Sep. 2019, pp. 481–486, doi: 10.1109/ICDAR.2019.00083.

[8] J. Zdenek and H. Nakayama, "JokerGAN: Memory-Efficient Model for Handwritten Text Generation with Text Line Awareness," in *MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia*, Oct. 2021, pp. 5655–5663, doi: 10.1145/3474085.3475713.

[9] J. Gan and W. Wang, "HiGAN: Handwriting Imitation Conditioned on Arbitrary-Length Texts and Disentangled Styles," in *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 9A, no. 9, pp. 7484–7492, May 2021, doi: 10.1609/aaai.v35i9.16917.

[10] A. K. Bhunia, S. Khan, H. Cholakkal, R. M. Anwer, F. S. Khan, and M. Shah, "Handwriting Transformers," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2021, pp. 1066–1074, doi: 10.1109/ICCV48922.2021.00112.

[11] N. Sasipriyaa, P. Natesan, E. Gothai, G. Madhesan, E. Madhumitha, and K. V. Mithun, "Recognition of Tamil handwritten characters using Scrabble GAN," in *2023 International Conference on Computer Communication and Informatics, ICCCI 2023*, Jan. 2023, pp. 1–5, doi: 10.1109/ICCCI56745.2023.10128564.

[12] Y. X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-Shot Learning from Imaginary Data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7278–7286, doi: 10.1109/CVPR.2018.00760.

[13] A. K. Bhunia, A. Das, A. K. Bhunia, P. S. R. Kishore, and P. P. Roy, "Handwriting recognition in low-resource scripts using adversarial learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2019, vol. 2019, pp. 4762–4771, doi: 10.1109/CVPR.2019.00490.

[14] P. Krishnan, K. Dutta, and C. V. Jawahar, "Word spotting and recognition using deep embedding," in *Proceedings - 13th IAPR International Workshop on Document Analysis Systems, DAS 2018*, Apr. 2018, pp. 1–6, doi: 10.1109/DAS.2018.70.

[15] 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, p. 100004, Apr. 2021, doi: 10.1016/j.jjimei.2020.100004.

[16] R. D. Urs and H. K. Chethan, "A Study on Identification and Cleaning of Struck-Out Words in Handwritten Documents," *Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2020*, 2021, pp. 87–95, doi: 10.1007/978-981-15-8530-2_6.

[17] U. V. Marti and H. Bunke, "The IAM-database: An English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, pp. 39–46, Nov. 2003, doi: 10.1007/s100320200071.

[18] E. Grosicki and H. El Abed, "ICDAR 2009 handwriting recognition competition," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009, pp. 1398–1402, doi: 10.1109/ICDAR.2009.184.

[19] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "CVL-database: An off-line database for writer retrieval, writer identification and word spotting," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, Aug. 2013, pp.

560–564, doi: 10.1109/ICDAR.2013.117.
[20] R. Manmatha and N. Srimal, "Scale space technique for word segmentation in handwritten documents," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1682, 1999, pp. 22–33, doi: 10.1007/3-540-48236-9_3.
[21] K. Lata, M. Dave, and N. K.N., "Data Augmentation Using Generative Adversarial Network," in *SSRN Electronic Journal*, 2019, doi: 10.2139/ssrn.3349576.
[22] L. Lan *et al.*, "Generative Adversarial Networks and Its Applications in Biomedical Informatics," *Frontiers in Public Health*, vol. 8, May 2020, doi: 10.3389/fpubh.2020.00164.
[23] P. Shukla, R. Aluvalu, S. Gite, and U. V. Maheswari, *Computer vision : applications of visual AI and image processing*, De Gruyter, 2023.
[24] O. N. Oyelade, A. E. Ezugwu, M. S. Almutairi, A. K. Saha, L. Abualigah, and H. Chiroma, "A generative adversarial network for synthetization of regions of interest based on digital mammograms," *Scientific Reports*, vol. 12, Apr. 2022, doi: 10.1038/s41598-022-09929-9.
[25] V. Khrulkov and I. Oseledets, "Geometry score: A method for comparing generative adversarial networks," in *35th International Conference on Machine Learning, ICML 2018*, vol. 6, pp. 4114–4122, 2018.
[26] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6629-6640, doi: 10.5555/3295222.3295408.

BIOGRAPHIES OF AUTHORS

Dheemanth Urs Raje received B.Eng. degree in Computer Science and Engineering from Visvesvaraya Technological University, M.Tech. degree in Computer Science and Technology from University of Mysore, and pursuing Ph.D. from Visvesvaraya Technological University. Currently. He is a Research Scholar at the Department of Computer Science and Engineering, Maharaja Institute of Technology, Visveswaraya Technological University. His research interests include digital image processing, pattern recognition, document analysis, handwritten text recognition, and optical character recognition. He can be contacted at email: dheemu.urs@gmail.com.

Chethan Hasigala Krishanappa completed Ph.D. from University of Mysore in the field of Digital Image Processing. Four of his students are pursuing Ph.D. under his guidance. His major area of research is digital image processing. He is currently working as Professor in Department of Computer Science and Engineering at Maharaja Institute of Technology, affiliated with Visvesvaraya Technological University. He has published numerous papers in recognized journals and conferences proceedings. His research interests include document analysis, handwritten recognition, and pattern recognition. He can be contacted at email: chethanhk@mitmysore.in.