



ARTICLE

A Harmonic Approach to Handwriting Style Synthesis Using Deep Learning

Mahatir Ahmed Tusher¹, Saket Choudary Kongara¹, Sagar Dhanraj Pande², SeongKi Kim^{3,*} and Salil Bharany^{4,*}

¹School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, 522237, India

²School of Engineering & Technology, Pimpri Chinchwad University, Pune, Maharashtra, 412106, India

³Department of Computer Engineering, Chosun University, Gwangju, 61452, Republic of Korea

⁴Independent Researcher, Amritsar, Punjab, 143001, India

*Corresponding Authors: SeongKi Kim. Email: skkim9226@gmail.com; Salil Bharany. Email: salil.bharany@gmail.com

Received: 25 December 2023 Accepted: 20 March 2024 Published: 20 June 2024

ABSTRACT

The challenging task of handwriting style synthesis requires capturing the individuality and diversity of human handwriting. The majority of currently available methods use either a generative adversarial network (GAN) or a recurrent neural network (RNN) to generate new handwriting styles. This is why these techniques frequently fall short of producing diverse and realistic text pictures, particularly for terms that are not commonly used. To resolve that, this research proposes a novel deep learning model that consists of a style encoder and a text generator to synthesize different handwriting styles. This network excels in generating conditional text by extracting style vectors from a series of style images. The model performs admirably on a range of handwriting synthesis tasks, including the production of text that is out-of-vocabulary. It works more effectively than previous approaches by displaying lower values on key Generative Adversarial Network evaluation metrics, such as Geometric Score (GS) (3.21×10^{-5}) and Fréchet Inception Distance (FID) (8.75), as well as text recognition metrics, like Character Error Rate (CER) and Word Error Rate (WER). A thorough component analysis revealed the steady improvement in image production quality, highlighting the importance of specific handwriting styles. Applicable fields include digital forensics, creative writing, and document security.

KEYWORDS

Recurrent neural network; generative adversarial network; style encoder; fréchet inception distance; geometric score; character error rate; mixture density network; word error rate

1 Introduction

Generating realistic handwritten text images corresponding to a particular style is known as “handwriting style synthesis”. This is a challenging task, especially for text that is arbitrary in length and out of the accessible vocabulary samples or contains invisible characters. Most existing approaches use either generative adversarial network (GAN) or recurrent neural network (RNN) to create new handwriting styles. Deep learning models like GAN and RNN can learn from data. While Generative Adversarial Networks can generate handwritten text images of acceptable quality, they frequently have



difficulty capturing the subtle variances and delicate nuances of the writing style. RNNs can produce the text's stroke order, although frequently resulting in distorted and fuzzy visuals. Individually, neither model can bring out the expected outputs. This paper presents a novel approach to writing new handwriting styles to address these problems. Two primary components have been chosen to build this method: One is the style encoder, and the other is the text generator. Generally, a style encoder is a type of deep learning model (RNN) that can be trained to extract a set of style images and a short style vector from them. After that, the text generator may employ this style vector to help generate the appropriate style [1]. Text generators are another sort of deep learning model that can generate high-quality text images individually, such as bidirectional and auto-regressive transformers (BART) [2]. This approach combines the advantages of RNNs and GANs. GAN creates a realistic image of the text in a specific style, and RNN creates the stroke sequence that forms the text. The approach can be applied in various fields, including digital forensics, document security, and creative writing. It can also be used to create custom signatures, seals, or stamps to authenticate papers and stop forgeries. It can also be used to produce artistic and expressive texts that convey the writer's mood and personality, such as letters, stories, and poems. Additionally, it can be used to examine and contrast handwriting styles, including pressures, slants, and strokes, which can disclose a writer's identity or traits. The application of the system to style transfer, style interpolation, and style alteration tasks is demonstrated in this study.

The architecture uses two discriminators for individual characters and their relationships, simulating human learning by considering both the larger context and minute details. A novel framework is suggested that performs better than the current methods in several areas. The following features of the system allow it to generate various and accurate handwriting representations. Initially, the handwriting styles are described by low-dimensional vectors, which can help the generator create the right style. These latent vectors can be used to create new styles that are not present in the dataset. This work essentially suggests an innovative approach for synthesizing handwriting styles. The key contributions are:

1. This study presents a hybrid model that combines the advantages of GANs for image quality and RNNs for stroke sequence accuracy, which effectively addresses difficulties in realistic image capture and stroke sequence generation.
2. A style encoder (RNN) and text generator are combined to produce a variety of handwriting styles.
3. Printed style images are used for encoding to develop new handwriting styles.
4. The use of low-dimensional vectors facilitates the creation of new styles as well as accurate style reproduction.
5. Dual discriminators are used to generate language that is more realistic and diversified.
6. By using printed style images, this proposed method can handle any text content, even outside of the dataset, by changing the wordings of the images.

These contributions extend beyond handwriting style synthesis, potentially impacting diverse fields. Creating and analyzing varied handwriting styles in digital forensics can aid in authentication and verification. Innovative methods to prevent forgery are offered for document security. In creative writing, authors can utilize the tool to explore new styles and expressions. Fig. 1 shows the proposed method's architecture overview and a few of the created words.

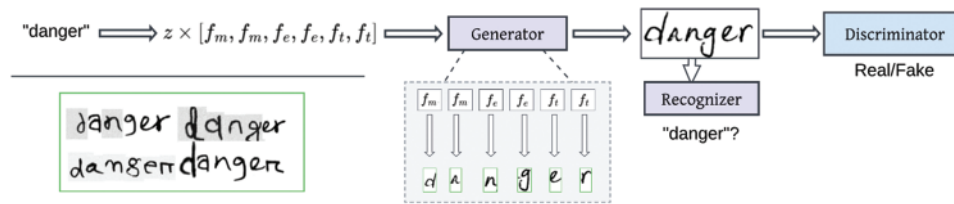


Figure 1: Architecture overview of the proposed approach's framework along with the generation of the word "danger"

Section 2 reviews the relevant works, and Section 3 presents the proposed architecture and its elements. Section 4 provides an overview of the approach, and Section 5 exhibits the results of experiments evaluating the framework's suitability for handwriting style synthesis. The study concludes in Section 6.

2 Literature Survey

An extensive survey of the literature was done to investigate relevant publications that served as inspiration for the current methodology. In order to improve text recognition using bidirectional LSTM recurrent layers, Alonso et al. [3] used generative adversarial network (GAN) to create artificial handwritten word pictures. With the use of a GAN-based system, Gan et al. [4] made a substantial contribution to handwriting synthesis by showing how to extract a variety of text and calligraphic styles without sacrificing visual appeal. Styles for arbitrary-Length and Out-of-vocabulary text based on a Generative Adversarial Network (SLOGAN) [5] by Luo et al. and the suggested method are similar in that they both use GANs to generate handwriting images with random vocabularies and styles. A semi-supervised approach to synthesize handwritten text images (ScrabbleGAN) was used to actual handwritten textual pictures by Fogel et al. [6], enhancing Handwritten Text Recognition (HTR). Graves [7] used recurrent neural networks with long short-term memory (LSTM) to produce realistic cursive handwriting synthesis. Wang et al. [8] highlighted the benefits of transfer learning in speech and language processing. In order to improve handwritten word recognition using synthetic datasets, Akter et al. provided a deep-text-recognition-benchmark (BiLSTM-CTC) based technique that shows promise in tackling data scarcity, especially for languages like Bangla [9]. A fresh assessment metric was introduced by Tüselmann et al. [10] to solve problems in semantic word recognition. The effect of orthographic neighbourhoods on word recognition was investigated by Grainger et al. [11].

A bidirectional attention and gated graph convolutional network were presented by Pande et al. [12] for text classification. In order to synthesize various handwritten text pictures, Liu et al. [13] devised a model that disentangles style and content. This model demonstrated superior performance in data augmentation and improved text recognition. By genuinely displaying a variety of handwritten word images with calligraphic style and literary substance, Kang et al. innovated image production and demonstrated breakthroughs through assessments [14]. The use of online handwriting analysis for Parkinson's disease diagnosis was investigated by Aouraghe et al. [15]. A supervised ConvNet was used by Pippi et al. [16] to improve handwriting analysis. GAN-based models capturing distinct handwriting styles were proposed by Kalingeri et al. [17] for use in font development. In the field of image-text fusion, Huang et al. presented a global-local fusion module for adversarial machine learning that improves accuracy and robustness by dynamically combining global and local similarity metrics [18]. By using multi-scale feature extraction and fusion techniques, Lu et al. improved the accuracy of Visual Question Answering (VQA) for better image and text representation [19].

The literature review, in summary, emphasises the variety of sources of inspiration for handwriting synthesis. Notably, Alonso et al.'s work improved text recognition by using GANs to create artificial handwritten word pictures. While Fogel et al. used ScrabbleGAN to improve HTR with text style adjustment, Luo et al. created SLOGAN for flexible handwriting production. The function of LSTM in realistic cursive handwriting synthesis was illustrated by Graves. Using innovative techniques, a number of studies tackled text detection, character classification, and semantic word recognition that demonstrate the continuous advancement and creativity in handwriting analysis.

3 Proposed Framework

This section of the study has covered the suggested method's framework. The two components that constitute the approach are a text generator and a style encoder. A convolutional neural network (CNN) is used as the style encoder to take a set of style images and create a low dimensional style vector. A recurrent neural network (RNN) serves as the text generator, sequentially generating high-quality text images.

3.1 Text Generator

A style vector and a printed style image are the inputs used by the text generator, a modified Sketch-RNN model, to create realistic text stroke by stroke. With the help of a convolutional layer and an attention mechanism, decoder, and encoder the model converts the printed style image into a feature map. The decoder produces strokes by using long short-term memory (LSTM) cells [20]. Soft attention, conditioned by a style vector, is required for decoding. By using normal distributions, a mixed density network can forecast stroke points [21]. Reconstruction and adversarial losses are measured during training, together with the generator's capacity to fool a discriminator and negative log-likelihood. The model is trained using 64 batches, 100 epochs per dataset, a learning rate of 0.0001, and data augmentation techniques like rotation, scaling, noise injection, and random cropping. The diagram of style vector (RNN sketch) has been shown in Fig. 2. The text generator as a function can formally be defined that maps the input style vector and text image to the output text image, as follows:

$$T: V_s \times I_t \rightarrow I_g \quad (1)$$

Here, T is the text generator function, V_s is the set of style vectors, I_t is the set of text images, and I_g is the set of generated text images.

3.2 Style Encoder

This study involves training a style encoder on a set of style images to learn the latent representation of handwriting styles. Each style image, which is an excerpt from a certain author's work, is unique. The encoder obtains a style vector by analyzing many style images from the same source. The text generator uses this style vector to produce text images with a consistent appearance. A 128-dimensional style vector is eventually generated by the encoder, which has adjusted certain model components, through learning by grouping related style vectors and isolating dissimilar ones [22]. Fig. 3 shows the diagram of 18 layered style encoder. The style encoder can formally be defined as a function that maps the input style image to the output style vector, as follows:

$$S: I_s \rightarrow V_s \quad (2)$$

Here, this S is the style encoder function, V_s is the set of style vectors, and I_s is the collection of style images.

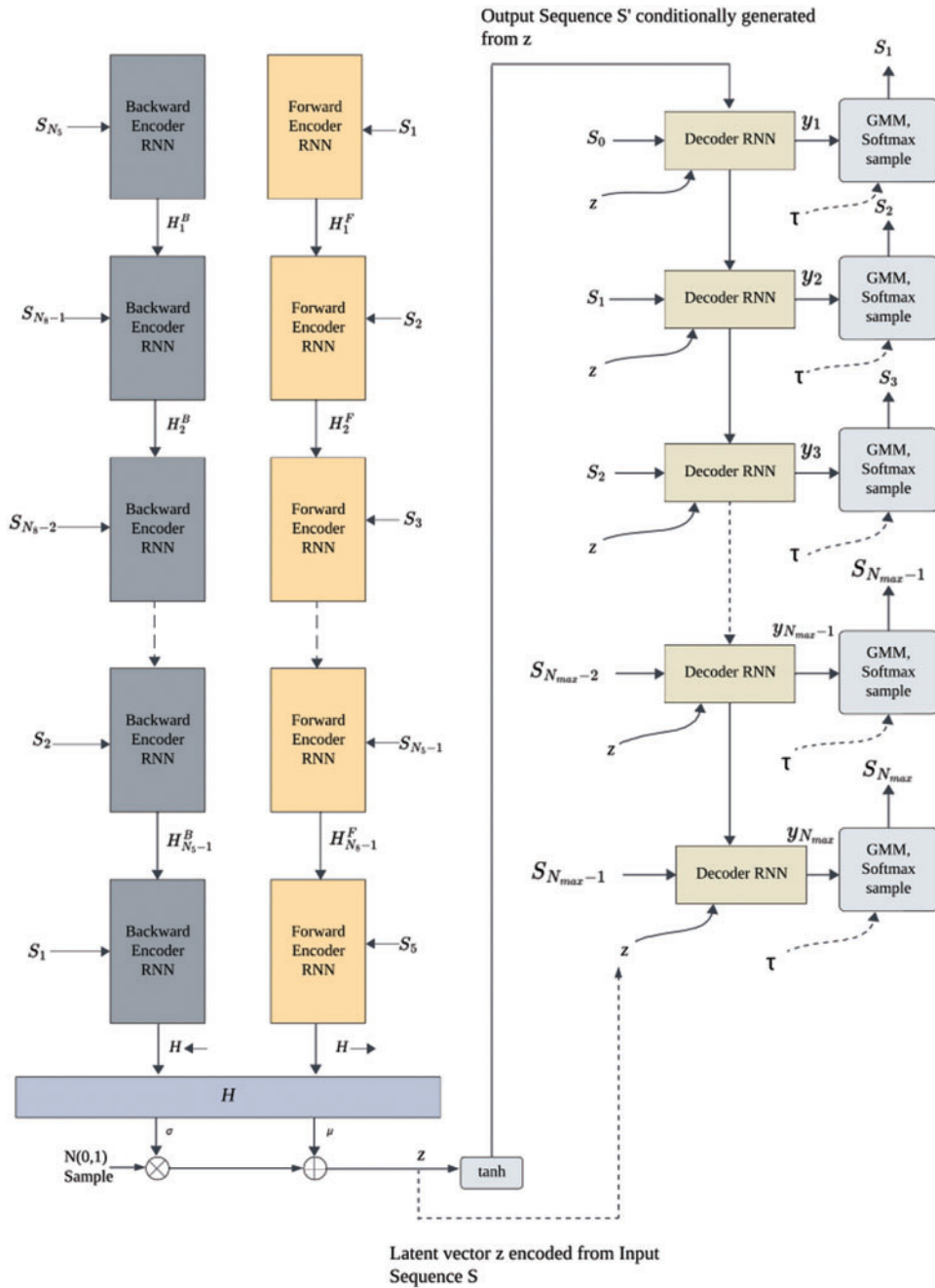


Figure 2: This sketch-recurrent neural network model is used to modify the text generator

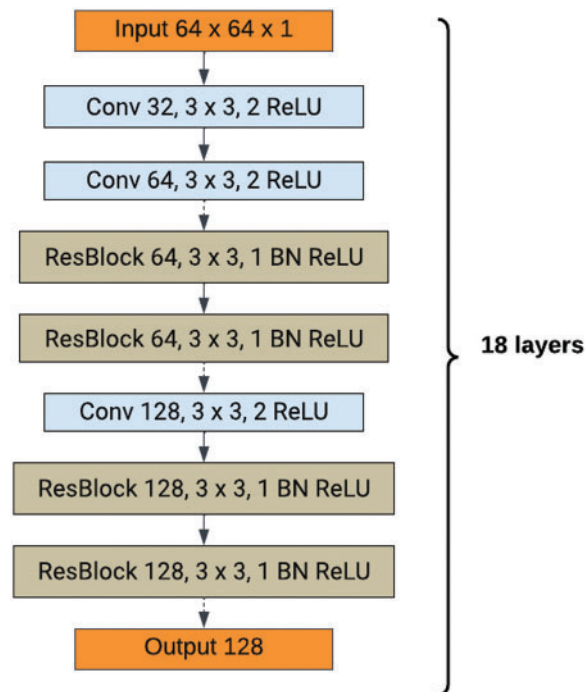


Figure 3: A 128-dimensional style vector

4 Methodology

The main topics and steps of the proposed framework for handwriting style synthesis are covered by this method. It comes with a high degree of style and content flexibility and numerous key stages that facilitate the creation of realistic and varied handwritten text graphics. Fig. 4 demonstrates the simplified architecture of the proposed method.

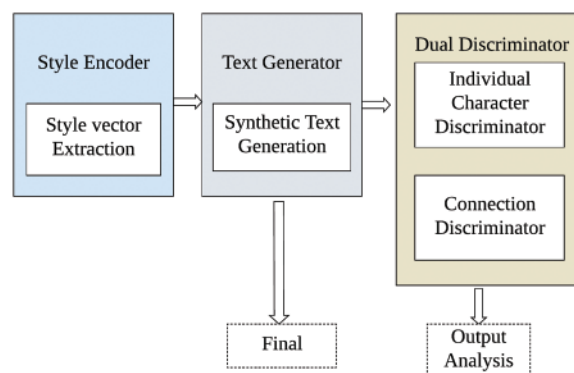


Figure 4: Model architecture for handwriting style synthesis with style encoder, text generator, and dual discriminator

4.1 Datasets

An extensive variety of text content and handwriting styles are represented in the chosen datasets for the study. The Integrated Argument Mining (IAM) Dataset (more than 115,000 English words from 657 authors) [23] and the Reconnaissance & Indexation de données Manuscrites et de fac Similes (RIMES) Dataset (more than 60,000 French words from 1,000 writers) [24] are two examples of the specially selected datasets that are used to train and assess the proposed handwriting style synthesis framework.

4.2 Handwriting Style Representation

The representation of handwriting styles forms the central component of the proposed methodology.

4.2.1 Style Bank and Synthesizer

Text Synthesiser and Style Bank are used by the suggested framework to generate text visuals. A lookup table called the Style Bank holds n latent vectors ($Z_{\text{all}} \in \mathbb{R}^{d \times n}$) representing different handwriting styles. The intended handwriting style is generated by the Text Synthesiser, which is directed by the latent vector Z that was acquired from the Style Bank. During training, the procedure entails updating the Text Synthesiser and Style Bank simultaneously. The equations below describe how the Text Synthesiser's encoder-decoder design converts an input printed character image (I_{print}) into an output image (I_{fake}):

$$Z = \text{StyleBank}(\text{writer feature vector}) \quad (3)$$

$$I_{\text{fake}} = \text{TextSynthesizer}(I_{\text{print}}, Z) \quad (4)$$

4.3 Dual Discriminators

The significance of cursive joins in handwritten text images is recognized and the Cursive Specific Discriminator (D_{join}) is introduced.

4.3.1 Character Segmented Discriminator (D_{char})

The Character Segmented Discriminator (D_{char}) was introduced to improve image quality and reduce under-fitting. It works at the character level. It employs an attention method and consists of $D_{\text{char}, \text{adv}}$ for adversarial training and $D_{\text{char}, \text{content}}$ for character content monitoring. While $D_{\text{char}, \text{content}}$ localises characters with text string labels as weak supervision, $D_{\text{char}, \text{adv}}$ makes adversarial training easier by sharing hidden states [25]. The Character Segmented Discriminator (D_{char}) mathematical equation is expressed as follows:

$$L_{D_{\text{char}}} = L_{D_{\text{char}, \text{adv}}} + \alpha L_{D_{\text{char}, \text{content}}} \quad (5)$$

Here, z = latent vector, I_{print} = printed image, I_{real} = real image, G = Text Synthesizer, and $D_{\text{char}, \text{adv}}$ = binary classifier [26]. The character content loss is defined as:

$$L_{D_{\text{char}, \text{content}}} = -E_{I_{\text{real}}}, Y \left[\sum_t y_t \log p(y_t | I_{\text{real}}, s_t) \right] \quad (6)$$

Y has been used to show the text name of the image, y_t has been used to show the one-hot code of the t -th letter, $p(y_t | I_{\text{real}}, s_t)$ to show the softmax result of $D_{\text{char}, \text{content}}$, and s_t to show the hidden state

of $D_{char, content}$.

$$a_t = softmax(W_t tanh(W_1 F + W_s s_t)) \quad (7)$$

$$c_t = \sum_{i=1}^N a_{t,i} f_i \quad (8)$$

$$S_t = LSTM(c_t, s_{t-1}) \quad (9)$$

Here, $F = [f_1, f_2, \dots, f_n]$ is the feature map of the image, W_1 , W_2 , and W_s are learnable weights, $a_t = [a_1, a_2, \dots, a_n]$ is the attention vector and the context vector is denoted by c_t . In (8), N is the number of input vectors x_i , and $a_{t,i}$ is the i -th element of the attention vector a_t . Therefore, the range of i here is from 1 to N , which means that, the attention mechanism sums over all the input vectors x_i to obtain the context vector c_t [27].

4.3.2 Cursive Specific Discriminator (D_{join})

Cursive Specific Discriminator (D_{join}) is a global discriminator developed to identify cursive joins in handwritten text images by evaluating the associations between neighbouring characters. It concentrates on overseeing the synthesizer and estimating particular handwriting styles at the cursive join level. The core of this method is the adversarial loss at the cursive join level. For D_{join} , the equation is:

$$L_{D_{join}} = L_{D_{join,adv}} + \alpha L_{D_{join,content}} \quad (10)$$

where $L_{D_{join}}$ is the total loss of D_{join} , $L_{D_{join,adv}}$ is the adversarial loss of $D_{join,adv}$, $L_{D_{join,ID}}$ is the handwriting style loss of $D_{join, ID}$, and the two losses are balanced by β , a hyperparameter. The adversarial loss of $D_{join,adv}$ is defined as:

$$L_{D_{join,adv}} = E_{I_{real}, z} [\log D_{join,adv}(I_{real}, z)] + E_{I_{print}, z} [\log(1 - D_{join,adv}(G(I_{print}, z), z))] \quad (11)$$

Here, I_{real} is the real image of a given handwriting style, z is the latent vector of the same handwriting style from the style bank, I_{print} is the printed character image with the same content as I_{real} , G is the Text Synthesizer, and $D_{join,adv}$ is a binary classifier that the likelihood that an image is authentic or fraudulent. Definition of $D_{join, ID}$'s handwriting style loss:

$$L_{D_{join,ID}} = -E_{I_{real}, z} [\log p(z|I_{real})] \quad (12)$$

where $p(z|I_{real})$ is the softmax output of $D_{join, ID}$ that predicts the handwriting style vector given an image [28].

4.4 Methodological Strategy and Algorithmic Representation

The algorithm of this study used a structured technique to build task-specific deep learning models and algorithms. In addition to providing explicit and unambiguous instructions for addressing the problem and describing the code used in experimental procedures, algorithm descriptions are given to highlight important elements and processes. Algorithm 1 has been demonstrated.

Algorithm 1: Algorithm for Enhanced Handwritten Character Generation Training Using Deep Learning

Input: G, L_idt, L_join_adv, style_bank, D_char, D_join, L_join_ID, L_char_content, optimizer_style_bank, L_char_adv, optimizer_G, optimizer_D_char, optimizer_D_join, num_iterations

(Continued)

Algorithm 1 (continued)**1: Initialization**

Set learning rate: Learning_rate=0.0002

Initialize networks and optimizers:

-G, D_{char}, D_{join}, style_bank

-optimizer_G, optimizer_D_char, optimizer_D_join, optimizer_style_bank

2: Training Loop**a. Get data batch**

-I_{print}, Y_{print}, I_{real}, Y_{real}, writer_ID

b. Update D_{char} and D_{join}

-Zero gradients: $\nabla D_{char} = 0, \nabla D_{join} = 0$

-style_latent \sim style_bank.sample()

-Calculate losses:

$L_{char_adv} = -\log(D_{char}(I_{real}, Y_{real}))$

$L_{join_adv} = -\log(D_{join}(I_{real}, Y_{real}, writer_ID))$

Backpropagate and update D_{char}, D_{join}

c. Update G and style_bank

-Zero gradients: $\nabla G=0, \nabla style_bank=0$

-style_latent \sim style_bank.sample()

-Calculate losses:

$L_{char_content} = \text{content_loss}(G(I_{print}, style_latent), I_{real})$

$L_{join_ID} = \text{ID_loss}(G(I_{print}, style_latent), Y_{real}, writer_ID)$

Backpropagate and update G, style_bank

d. Print and save model checkpoints

-Every 100 iterations

e. Generate samples for evaluation

-Every 500 iterations

f. Adjust learning rate

-learning_rateG=adjust_learning_rate(optimizer_G, epochs)

3: Return trained networks

style_bank, G, D_{join}, D_{char}

4: End**5 Result and Analysis**

The architecture and implementation of the model are covered in this section. The generator and two discriminators in the model were trained using various loss functions. The dataset, network, optimisation, and measurements are all part of the implementation. The overview of network architecture, performance metrics and the system configuration for testing and training are displayed in [Table 1](#).






Table 1: Summary of system configuration for model training and testing

Aspect	Details
Hardware	i9-12900K (16 cores), RTX 4090 (48 GB VRAM), 64 GB RAM, 2 TB NVMe SSD
Software	Windows 11, TensorFlow 3.0, Python 3.10.2, CUDA 12.0
Training	5 days, Preprocessing: Augmentation, resizing, normalization, 1.5 M images, 50,000 validation
Network	Generator: 6 blocks, 5 conv layers, 5 deconv layers, Style Vector: 256
Training parameters	Optimizer: Adaptive Moment, $\beta_1: 0.7$, $\beta_2: 0.995$, LR reduction 5×10^{-4} to 5×10^{-5} at 250 K iterations, Batch: 64
Image processing	Scaling: Preserve aspect ratio, height 128 px, Padding: White < 300 px, resize to 500 px if wider

5.1 Component Analysis

The method looks at every element and how it affects the other elements in detail in order to demonstrate how effective the recommended elements are. [Table 2](#) summarizes the analysis's incremental results using the RIMES dataset and provides a step-by-step method for assessing the efficacy of the suggested elements. A thorough breakdown of the component analysis is given by the visualizations in [Fig. 4](#), which feature a parallel line chart to display the relevant Geometric Scores and a bar graph to indicate how different components affect Frechet Inception Distance (FID) scores.

Table 2: The component analysis result

Components included	FID	GS	Picture (“and”)
$L_{\text{join, adv}}$	180.50	3.00×10^{-2}	
L_{id}	38.90	5.21×10^{-3}	
$L_{\text{char, content}}$	12.50	7.01×10^{-4}	
$L_{\text{char, adv}}$	10.90	6.04×10^{-4}	
Specific handwriting styles	8.21	3.21×10^{-5}	

5.1.1 Baseline and Identical Mapping Loss

The baseline, which uses only the $L_{\text{join, adv}}$ component, generates a framework that is similar to an adversarial loss PatchGAN, except it uses a noise vector “n” rather of the latent vector “z.” This draws attention to restrictions, which are demonstrated by absurd strokes and a collapse state denoted by noticeably elevated FID and GS scores ([Table 2](#)). When the same mapping loss, L_{id} , is introduced, stability and effectiveness are improved, and distinct image features appear in comparison to the baseline. One way to support text content preservation is to include character-level content loss,

Lchar, content. Fig. 5 illustrates the crucial function of Identical Mapping Loss (L_{id}) and its temporal progression.

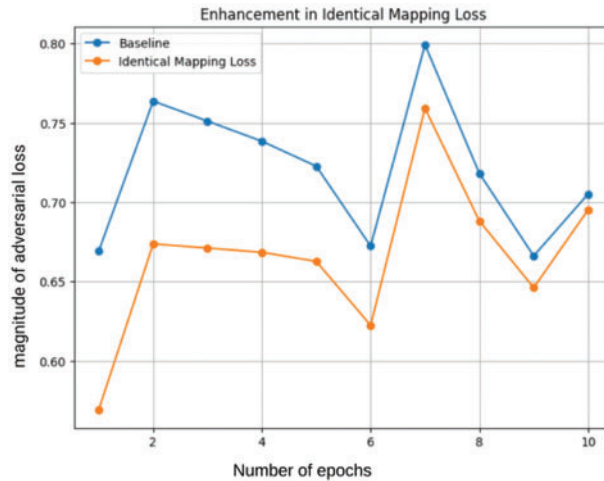


Figure 5: Temporal progression of mapping loss with the L_{id} component introduction across training epochs, with x-axis denoting epochs (iterations) and y-axis showing adversarial loss magnitude

5.1.2 Character-Level Adversarial Loss

The character-level adversarial loss, $L_{char, adv}$, is introduced to further enhance the realism of individual characters. At this stage, the generated images exhibit remarkable realism and quality. This component underscores the critical role of dual discriminators in facilitating high-quality handwritten text image generation. A graph of adversarial loss has been shown in Fig. 6. A few more graphs of loss finctions over iterations or epochs during the training phase has been demonstrated in Fig. 7. These graphs have shown the loss values of the generator, the discriminator, and the classifier of the generative adversarial network (GAN) during the training process.

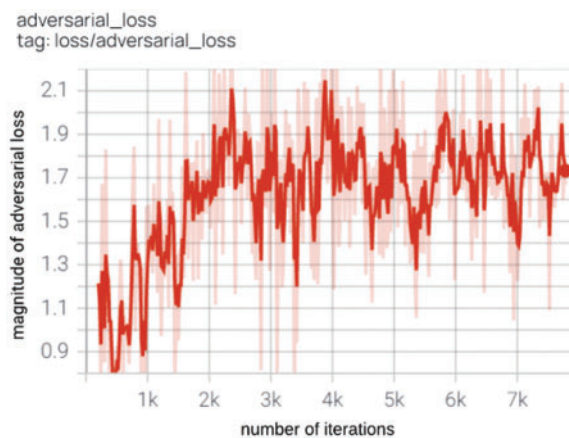


Figure 6: Adversarial loss during the trainig phase where a correlation exists between adversarial loss (0.9–2.1) on the y-axis and the x-axis (iterations: 0–7,000)

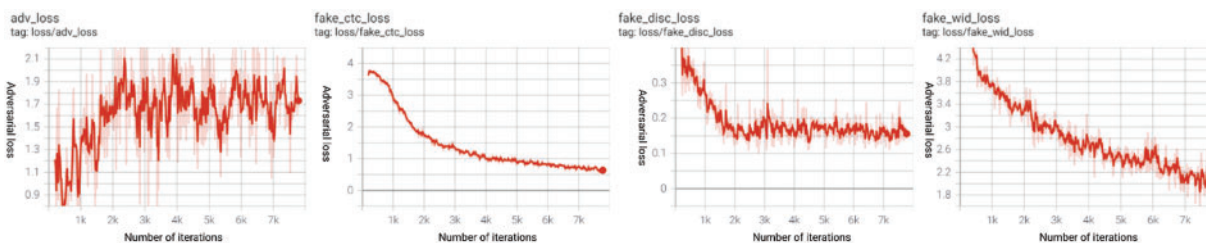


Figure 7: GAN component losses during training. x-axis for iterations, y-axis for adversarial loss magnitude

5.1.3 Specific Handwriting Styles

Once the component analysis reaches its pinnacle, distinct, recognisable handwriting styles are mandated. The generator incorporates complicated features by utilising writer identifiers to guide it and preserving handwriting traits in the style bank. A low Geometric Score (GS) of 3.21×10^{-5} and Fréchet Inception Distance (FID) of 8.75 demonstrate the efficacy of the framework. The reference-guided results in Fig. 8 are presented in a particular handwriting style.

Specific style	Generated text
font	This is an example
font	This is an example
font	This is an example,
font	This is an example
font	This is an example
font	This is an example
font	This is an example
font	This is an example

Figure 8: For every handwriting style, eight sentences were created using a selected reference word

5.2 Generating Words That Are Out-of-Vocabulary

The ability of the system to generate concepts outside of the training language demonstrates its adaptability. This study confirms the effectiveness of the strategy even in the absence of handwriting-specific training examples. As illustrated in Fig. 9, a negative association is found between the Fréchet Inception Distance (FID) score and sample size, with higher output quality being correlated with lower FID ratings.

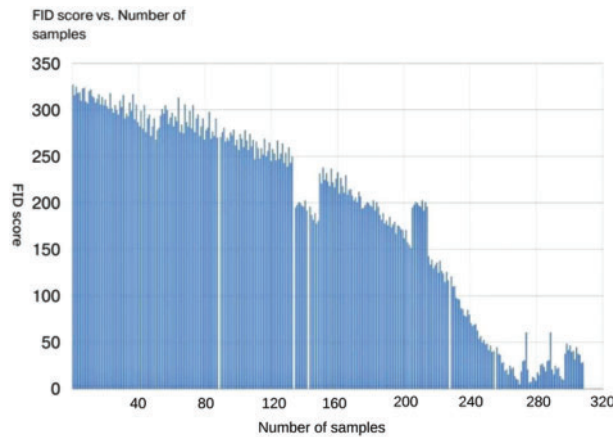


Figure 9: Fréchet inception distance score against number of samples in case of words that are out of vocabulary

[Table 3](#) provides thorough statistics that show how well the process creates terms that are not already in the existing lexicon.

Table 3: Comparison with prior studies in case of generating words out of vocabularies

Method	FID
Luo et al. [5]	97.81
Kang et al. [14]	125.87
Ours	97.03

5.3 Evaluation of the Approach against Prior Works

This section uses geometric score and Fréchet Inception Distance (FID) measurements to compare the generated images with models from previous works. Lower values suggest better outcomes. The assessment measures, FID and Geometric Score (GS), show how well the model performs. A graphical comparison with earlier research is shown in [Fig. 10](#), and a comparison of the suggested method's performance metrics with those of Alonso et al. [3], Gan et al. [4], Luo et al. [5], Fogel et al. [6], Akter et al. [9] and Kang et al. [14] is shown in [Table 4](#).

Furthermore, the handwriting synthesis model is evaluated using Word Error Rate (WER) and Character Error Rate (CER) assessment metrics, where lower values indicate higher performance, and compared with modern approaches (Alonso et al. [3], Fogel et al. [6], Luo et al. [5], Kang et al. [14]). The proposed approach outperformed previous studies in all metrics except for CER on the Reconnaissance & Indexation de données Manuscrites et de fac Similes (RIMES) dataset, as shown in [Table 5](#).

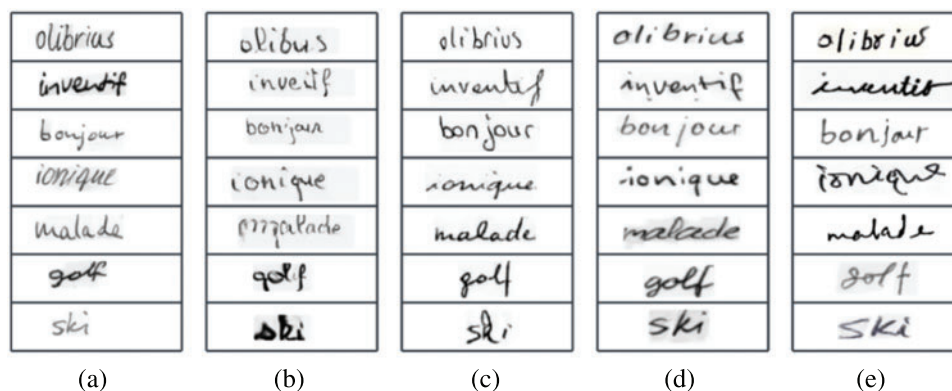


Figure 10: The comparison between the proposed method's result with following prior studies: Luo et al. [5] (a), Alonso et al. [3] (b), Fogel et al. [6] (c), and Gan et al. [4] (d) and (e) in the figure displays the proposed approach's result

Table 4: Comparison of model results (Geometric score and Fréchet inception distance score) with prior studies

Method	GS	FID
Alonso et al. [3]	8.58×10^{-4}	23.94
Gan et al. [4]	–	17.28
Luo et al. [5]	5.59×10^{-4}	12.06
Fogel et al. [6]	7.60×10^{-4}	23.78
Akter et al. [9]	0.98	–
Kang et al. [14]	–	120.07
Ours	3.21×10^{-5}	8.75

Table 5: Word error rate (WER) and character error rate (CER) of several handwriting synthesis models on RIMES and IAM datasets are compared

Studies	CER		WER	
	RIMES	IAM	RIMES	IAM
Kang et al. [14]	–	6.75	–	17.26
Gan et al. [4]	3.35	5.95	11.50	14.97
Grainger et al. [11]	3.57	13.42	11.32	23.61
Lewis et al. [2]	4.03	–	11.90	–
Ours	3.27	5.32	11.57	14.71

5.4 Diversity of Generation

With the help of its fully convolutional generator, the framework effectively manipulates input printed style pictures and latent style vectors to produce a variety of handwritten text images while maintaining spatial consistency [29,30]. This feature enables the production of graphics with various shaped text, modified letter spacing, and variable sentence lengths. Fig. 11 shows how the generator may be used to create a variety of styles by manipulating latent vectors.

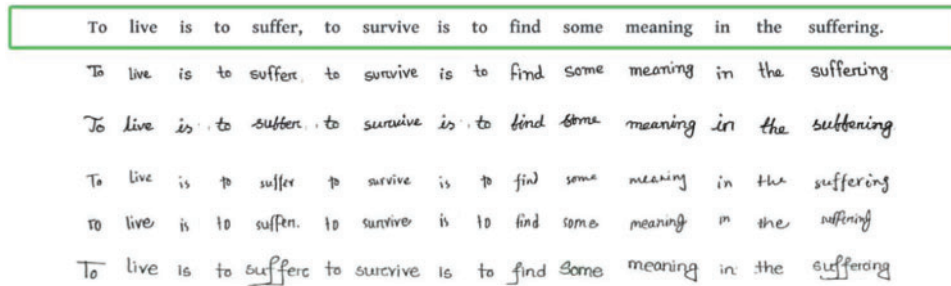


Figure 11: The diversity of the long text generation of the proposed architecture is demonstrated

The method also performs exceptionally well in style interpolation, combining various handwriting styles to create smooth transitions, as seen in Fig. 12 for words like “hope,” “Miracle,” “Life,” and “death.” These illustrations show the model’s deft handling of stylistic changes, demonstrating its versatility and dexterity across domains.

Style Interpolation	"hope"	"Miracle"	"Life"	"death"
style "x"	hope	Miracle	Life	death
↑	hope	Miracle	Life	death
↑	hope	Miracle	Life	death
↑	hope	Miracle	Life	death
↑	hope	Miracle	Life	death
↓	hope	Miracle	Life	death
style "y"	hope	Miracle	Life	death

Figure 12: Style interpolation outcomes for selected words (‘hope,’ ‘Miracle,’ ‘Life,’ ‘death’) in handwriting synthesis

5.5 Domain Adaptation

Table 6 displays domain adaptation experiments that demonstrate the adaptability of the model. Exclusive training set the initial benchmarks, starting with Integrated Argument Mining (IAM) as the baseline. The addition of synthetic styles significantly enhanced performance. The hybrid model, which combined the IAM and Reconnaissance & Indexation de données Manuscrites et de fac Similes (RIMES) datasets, showed a good trade-off, and the Oracle model performed best when trained alone on the RIMES dataset.

Table 6: The influence of handwriting synthesis on the IAM and RIMES datasets is demonstrated through performance measures for training configurations, model architectures, and domain adaption strategies that include word error rate (WER) and character error rate (CER) with standard deviations

Training data configuration	Model configuration	Domain adaptation	WER (%)	CER (%)
IAM only	Baseline model	No adaptation	40.00 ± 1.00	20.00 ± 0.50
IAM + Synthetic	Enhanced model	Synthetic styles	25.00 ± 0.50	12.50 ± 0.25
IAM + RIMES	Advanced model	Mixed styles	30.00 ± 0.75	15.00 ± 0.30
RIMES only	Oracle model	Optimal adaptation	22.50 ± 0.25	10.00 ± 0.20
IAM (ours)	Enhanced model	Mixed styles	14.71 ± 0.50	5.32 ± 0.25
RIMES (ours)	Enhanced model	Mixed styles	11.57 ± 0.50	3.27 ± 0.25

6 Conclusion and Future Scope

This innovative method provides a thorough process to generate realistic handwritten text images for a range of styles and goals. The use of style vectors, dual discriminators, and multidimensional loss functions significantly improves the generated samples' quality and diversity. In particular, the method excels in tasks like as text recognition, Generative Adversarial Network (GAN) assessment, creating diversity, and domain adaptability. It is more helpful in the text recognition and computer vision industries due to its ability to generate words that are not included in dictionaries and adapt to different handwriting styles. The integration of a diverse set of training data enhances the text recognizer's robustness and efficacy. The trials show a significant improvement in text recognizer performance, with a notable decrease in Word Error Rate (WER) with the synthesis of roughly 10 million samples. A move in the direction of maximum efficiency is shown by larger simulated instances. The results demonstrate the framework's potential applications in the fields of optical character recognition (OCR), data augmentation, and text generation. Further study endeavors may enhance the synthesis method even further and explore additional beneficial applications, like the revitalization and conservation of threatened or extinct scripts.

Acknowledgement: None.

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean government (MSIT) (NRF-2023R1A2C1005950).

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: M.A. Tusher, S.C. Kongara; data collection: S.C. Kongara; analysis and interpretation of results: S.D. Pande, S. Bharany, S. Kim; draft manuscript preparation: M.A. Tusher, S.D. Pande. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data from the authors are accessible upon request. The data that support the findings of this study are available from the corresponding author, SeongKi Kim, upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] Y. Zhang, Y. Zhang, and W. Cai, "Separating style and content for generalized style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 8387–8396.
- [2] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguist.*, 2020, pp. 7871–7880.
- [3] E. Alonso, B. Moysset, and R. Messina, "Adversarial generation of handwritten text images conditioned on sequences," in *Proc. Int. Conf. Doc. Anal. Recognit. (ICDAR)*, Sydney, NSW, Australia, 2019, pp. 481–486.
- [4] J. Gan and W. Wang, "HiGAN: Handwriting imitation conditioned on arbitrary-length texts and disentangled styles," in *Proc. AAAI Conf. Artif. Intell.*, Palo Alto, California, USA, vol. 35, no. 9, pp. 7484–7492, 2021. doi: [10.1609/aaai.v35i9.16917](https://doi.org/10.1609/aaai.v35i9.16917).
- [5] C. Luo, Y. Zhu, L. Jin, Z. Li, and D. Peng, "SLOGAN: Handwriting style synthesis for arbitrary-length and out-of-vocabulary text," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8503–8515, 2023. doi: [10.1109/TNNLS.2022.3151477](https://doi.org/10.1109/TNNLS.2022.3151477).
- [6] S. Fogel, H. Averbuch-Elor, S. Cohen, S. Mazor, and R. Litman, "ScrabbleGAN: Semi-supervised varying length handwritten text generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 4323–4332.
- [7] A. Graves, "Generating sequences with recurrent neural networks," arXiv preprint arXiv:1308.0850, 2013.
- [8] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *Proc. Asia-Pacific Signal Inform. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Hong Kong, China, 2015, pp. 1225–1237.
- [9] S. Akter, H. Shahriar, A. Cuzzocrea, N. Ahmed, and C. Leung, "Handwritten word recognition using deep learning approach: A novel way of generating handwritten words," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Osaka, Japan, 2022, pp. 5414–5423.
- [10] O. Tüselmann, F. Wolf, and G. A. Fink, "Identifying and tackling key challenges in semantic word spotting," in *Proc. 17th Int. Conf. Front. Handwriting Recognit. (ICFHR)*, Dortmund, Germany, 2020, pp. 55–60.
- [11] J. Grainger, J. K. O'regan, A. M. Jacobs, and J. Segui, "On the role of competing word units in visual word recognition: The neighborhood frequency effect," *Percept Psycho.*, vol. 45, no. 3, pp. 189–195, 1989. doi: [10.3758/BF03210696](https://doi.org/10.3758/BF03210696).
- [12] S. D. Pande, T. Kumaresan, G. R. Lanke, S. Degadwala, G. Dhiman and M. Soni, "Bidirectional attention mechanism-based deep learning model for text classification under natural language processing," in *Proc. Int. Conf. Intell. Comput. Netw. (ICICN)*, Mumbai, India, 2023, pp. 417–427.
- [13] X. Liu, G. Meng, S. Xiang, and C. Pan, "Handwritten text generation via disentangled representations," *IEEE Signal Process. Lett.*, vol. 28, pp. 1838–1842, 2021. doi: [10.1109/LSP.2021.3109541](https://doi.org/10.1109/LSP.2021.3109541).
- [14] L. Kang, P. Riba, Y. Wang, M. Rusiñol, A. Fornés and M. Villegas, "GANwriting: Content-conditioned generation of styled handwritten word image," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, UK, 2020, pp. 273–289.
- [15] I. Aouraghe, G. Khaissidi, and M. Mrabti, "A literature review of online handwriting analysis to detect Parkinson's disease at an early stage," *Multimed. Tools Appl.*, vol. 82, no. 9, pp. 11923–11948, 2023. doi: [10.1007/s11042-022-13759-2](https://doi.org/10.1007/s11042-022-13759-2).
- [16] V. Pippi, S. Cascianelli, L. Baraldi, and R. Cucchiara, "Evaluating synthetic pre-training for handwriting processing tasks," *Pattern Recognit. Lett.*, vol. 172, pp. 44–50, 2022. doi: [10.1016/j.patrec.2023.06.003](https://doi.org/10.1016/j.patrec.2023.06.003).
- [17] R. Kalingeri, V. Kushwaha, R. Kala, and G. C. Nandi, "Synthesis of human-inspired intelligent fonts using conditional-DCGAN," *Comput. Vis. Mach. Intell.*, vol. 586, pp. 619–630, 2023. doi: [10.1007/978-981-19-7867-8](https://doi.org/10.1007/978-981-19-7867-8).

- [18] S. Huang, W. Fu, Z. Zhang, and S. Liu, "Global-local fusion based on adversarial sample generation for image-text matching," *Inf. Fusion*, vol. 103, pp. 102084, 2024. doi: [10.1016/j.inffus.2023.102084](https://doi.org/10.1016/j.inffus.2023.102084).
- [19] S. Lu, Y. Ding, M. Liu, Z. Yin, L. Yin and W. Zheng, "Multiscale feature extraction and fusion of image and text in VQA," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 54, pp. 268, 2023. doi: [10.1007/s44196-023-00233-6](https://doi.org/10.1007/s44196-023-00233-6).
- [20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [21] M. Zhu, C. Chen, N. Wang, J. Tang, and C. Zhao, "Mixed attention dense network for sketch classification," *Appl. Intell.*, vol. 51, no. 10, pp. 7298–7305, 2021. doi: [10.1007/s10489-021-02211-x](https://doi.org/10.1007/s10489-021-02211-x).
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.
- [23] U. V. Marti and H. Bunke, "The IAM-database: An English sentence database for offline handwriting recognition," *Int. J. Doc. Anal. Recognit.*, vol. 5, pp. 39–46, 2002. doi: [10.1007/s100320200071](https://doi.org/10.1007/s100320200071).
- [24] E. Grosicki, M. Carré, J. M. Brodin, and E. Geoffrois, "Results of the RIMES evaluation campaign for handwritten mail processing," in *Proc. 10th Int. Conf. Doc. Anal. Recognit.*, Barcelona, Spain, 2009, pp. 941–945.
- [25] Y. Kong *et al.*, "Look closer to supervise better: One-shot font generation via component-based discriminator," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 13472–134812.
- [26] C. Luo, L. Jin, and Z. Sun, "MORAN: A multi-object rectified attention network for scene text recognition," *Pattern Recognit.*, vol. 90, pp. 109–118, 2019. doi: [10.1016/j.patcog.2019.01.020](https://doi.org/10.1016/j.patcog.2019.01.020).
- [27] J. Fan, H. Wang, Y. Huang, K. Zhang, and B. Zhao, "AEDmts: An attention-based encoder-decoder framework for multi-sensory time series analytic," *IEEE Access*, vol. 8, pp. 37406–37415, 2020. doi: [10.1109/ACCESS.2020.2971579](https://doi.org/10.1109/ACCESS.2020.2971579).
- [28] X. Tian, F. Yang, and F. Tang, "Generating structurally complete stylish chinese font based on semi-supervised model," *Appl. Sci.*, vol. 13, no. 19, pp. 10650, 2023. doi: [10.3390/app131910650](https://doi.org/10.3390/app131910650).
- [29] E. A. Adeniyi, P. B. Falola, M. S. Maashi, M. Aljebreen, and S. Bharany, "Secure sensitive data sharing using RSA and ElGamal cryptographic algorithms with hash functions," *Information*, vol. 13, no. 10, pp. 442, 2022. doi: [10.3390/info13100442](https://doi.org/10.3390/info13100442).
- [30] P. Karthik, P. Shanthibala, A. Bhardwaj, S. Bharany, H. Yu and Y. B. Zikria, "A novel subset-based polynomial design for enhancing the security of short message-digest with inflated avalanche and random responses," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 1, pp. 310–323, 2023. doi: [10.1016/j.jksuci.2022.12.002](https://doi.org/10.1016/j.jksuci.2022.12.002).