Enhancing GAN-Based Handwriting Generative Model: Handwriting Feature Extraction through LSTM and Transformer

121090272 Lu Li 121090711 Yiqu Yang
121090607 Jingxuan Wu 121090532 Caijun Wang

1 Introduction

Handwriting analysis is crucial for applications like signature authentication and document verification, benefiting and being challenged by the uniqueness of handwriting styles. Feature extraction, a key aspect in these applications, confronts two major challenges: the labor-intensive collection of diverse annotated handwriting datasets and the difficulty in representing individual variability in handwriting features such as character shape, stroke thickness, and slant.

To address these challenges, researchers commonly use the IAM dataset[1] for training generative models and employ CNN-based style encoders such as HiGAN+[2], TextStyleBrush[3], and GANwriting[4] to extract handwriting features.

Our study focuses on advancing handwriting feature extraction by developing an automated processing pipeline capable of rapidly processing large volumes of handwriting images to generate an automatically annotated dataset. Additionally, we are experimenting with innovative frameworks through LSTM and Transformer for the style encoder to enhance both accuracy and adaptability. These efforts are poised to advance the development of more effective handwriting analysis technologies.1 The source code for our implementations is available on GitHub at this repository.

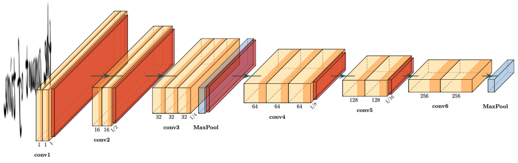

Figure 1: the structure of the CNN-based style encoder

2 Significance and Novelty

Our work holds significant importance in the realm of handwriting feature extraction. While the IAM dataset has been a valuable resource with 63,401 word-level images from 500 writers, our dataset, comprising 22,514 word-level images from 385 writers, represents a novel and reliable addition to the existing corpus. By merging the IAM dataset[1] with our own, we have created a more diverse and extensive dataset for training GAN models, addressing the limitations of data scarcity and enabling more comprehensive model training.

Another key contribution of our study lies in the exploration of RNN-based LSTM and the advanced Transformer structure for extracting content-independent features from handwriting. While LSTM has been extensively used in various sequential data analysis tasks, its success in handwriting feature extraction represents a novel and impactful finding. This success showcases the potential for LSTM to outperform more advanced structures in specific contexts. Our findings challenge conventional wisdom and pave the way for reevaluating the role of different methods in handwriting feature extraction, thereby contributing to the ongoing evolution of handwriting recognition technologies.

3 Data Collection Process

3.1 Dataset and Pipeline Overview

Our project has established an automated pipeline that significantly enriches the diversity and strengthens the robustness of our dataset:

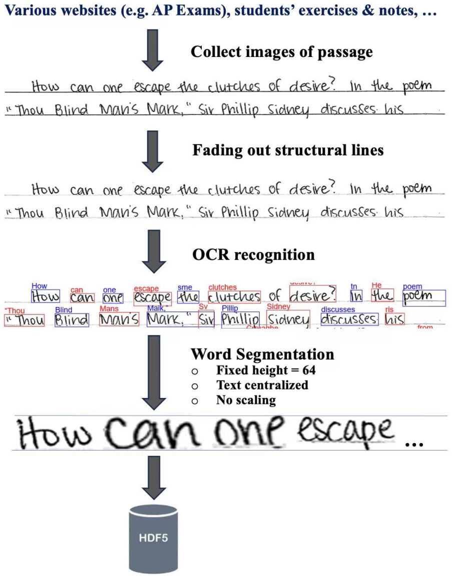
Various websites (e.g. AP Exams), students' exercises & notes, ...
Collect images of passage
How can one escape the clutches of desire? In the poem
"Thou Blind Man's Mark," Sir Phillip Sidney discusses his
Fading out structural lines
OCR recognition
Word Segmentation
Fixed height = 64
Text centralized
No scaling
How can one escape ...
HDF5

Figure 2: Data Collection Pipeline

- **Collection:** We collect a wide range of handwriting samples from various sources, including online documents and personal notes.
- **Preprocessing:** Structural lines within the images are faded to minimize noise prior to OCR processing.
- **OCR and Segmentation:** Our OCR engine identifies text bounding boxes and labels automatically. We then segment the preprocessed images into word-level images, ensuring that each word is centered and resized to a uniform height of 64 pixels without scaling.

3.2 Advantages and Challenges

This pipeline significantly automates key processing steps, facilitating the rapid preparation of extensive datasets crucial for advanced model training. The principal challenge lies in the manual corrections required to address OCR errors in intricate handwriting styles, which are essential to maintain the high quality of data necessary for the accuracy of model outcomes.

4 Methodology

We employed two classic architectures to recognize handwritten text in images: one based on the attention mechanism of the Transformer architecture, and the other based on the gating mechanism of the LSTM architecture. Each approach has its own strengths, combining different neural network advantages to better capture and analyze the features and styles of handwriting.

4.1 Transfomer

Our model architecture leverages the strengths of Convolutional Neural Networks (CNNs) and Transformers. Each segment extracted via a sliding window is first processed by dedicated convolutional layers for attention's queries (q), keys (k), and values (v). These layers extract local features and control feature quantity, preventing overfitting to local patterns while ensuring the capture of global features. We use a ConvEmbed layer to transform input images into a series of patches (embeddings). Each image is projected through a 2D convolutional layer to generate feature maps, which are rearranged into patches serving as input tokens for the Transformer. This initial embedding ensures effective capture of local spatial features.

The convolutional output is then fed into the Transformer, comprising multiple modules, each with an attention layer and a multi-layer perceptron (MLP). Each module normalizes its input, applies attention, adds residual connections, and processes the result through the MLP. This structure captures complex handwriting patterns and styles. Multiple attention heads enable the model to consider various handwriting aspects simultaneously, enhancing recognition capability. The entire model consists of multiple VisionTransformer stages, each with a series of Transformer blocks. These stages allow for hierarchical feature extraction, focusing on different abstraction levels. This layered structure ensures a comprehensive capture of intricate handwriting features and overall style. 3

4.2 Long Short-Term Memory

Additionally, we experimented with LSTM to achieve similar results. Traditional Long Short-Term Memory (LSTM) networks are designed for sequential data, making them suitable for processing handwriting sequences. LSTM units retain long-term dependencies and capture temporal patterns in handwriting strokes, crucial for style recognition. By feeding segmented image patches into an LSTM network, the model learns the sequential dependencies between these patches, effectively capturing handwriting style. Unlike traditional LSTM models that conclude with a fully connected layer to translate outputs, we omitted this layer to focus on the sequential nature of the data. Our LSTM processes the sequence of image patches, capturing temporal dynamics and the global context of handwriting style. Each LSTM cell helps understand the progression and flow of handwriting, enhancing recognition ability.

Our LSTM-based approach leverages the capability of LSTM networks to manage sequential dependencies and capture long-term patterns. By concentrating on the sequence of image patches, the LSTM learns the unique characteristics of handwriting styles. The output is a probability distribution over different handwriting styles, providing a probabilistic prediction for each style. This makes the LSTM a robust alternative to Transformer-based models for handwriting style recognition. 4

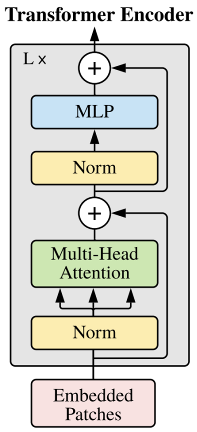

Figure 3: structure of the transformer encoder

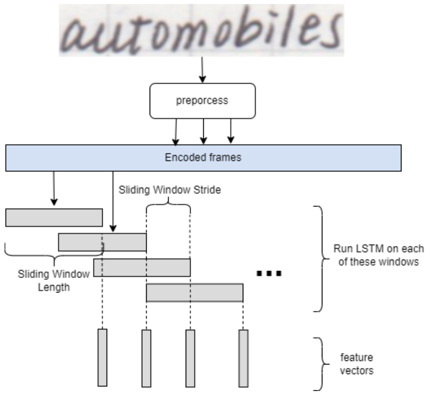

Figure 4: structure of the LSTM

5 Experiments and results

We first tested our feature extraction model on a writer classification task. We trained the writer classification model on one NVIDIA 3090 with batch size 256 and a base learning rate of 0.001, and the writer embedding hidden layer has dimension 256. We did three combinations of the writer identifier model and the handwriting dataset:

1. HiGAN+(2022)[2] + IAM dataset
2. HiGAN+(2022)[2] + IAM & Our dataset
3. Our Feature Extraction Model + IAM & Our dataset

The writer identifier loss is shown in figure 5, and the classification accuracy is shown in table 1. The experiment showed that our model converges slower compared to the original HiGAN+ writer identifier model but performs better and reached 91.15% accuracy, higher than the original HiGAN+[2] model trained on either the IAM dataset or our merged dataset.

Method	Accuracy
Our model on merged dataset	91.15%
Higan+ on merged dataset	90.07%
Higan+ on IAM dataset	87.08%

Table 1: Comparison of methods

Then, we plug our feature extraction module into a handwriting generation model to generate style-transferred handwriting images. We use the pipeline from HiGAN+[2] with our feature extraction module as style encoder and writer identifier. We first train our writer identifier and use a pre-trained OCR module for loss calculation. After that, we trained the GAN part with the pre-trained writer identifier and OCR module to get the discriminator and generator. The handwriting image generator model overview is shown in figure 6. E is the style encoder, using our well-trained style encoder model with LSTM; I is the writer identifier, which is trained based on the style encoder and is also pre-trained. G is the generator, and D is the discriminator of the GAN module. Finally, the objective evaluation of the generated image is calculated by module R, the OCR module, which evaluates how well the characters can be recognized.

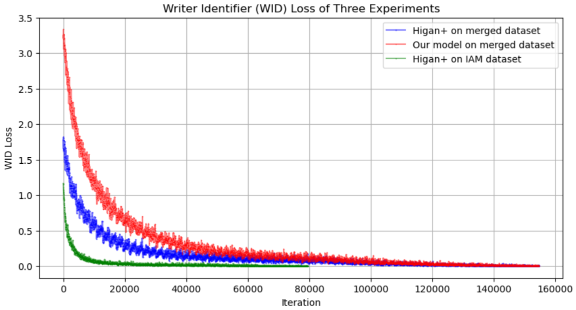

Figure 5: Writer classification loss

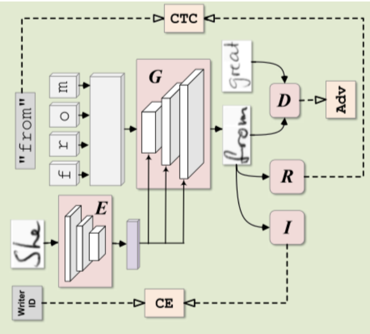

Figure 6: GAN module overview

We trained this model with one NVIDIA 3090, batch size 8, and the base learning rate of 10^{-5} and trained around 80k steps. In figure 7, some of the results are presented. The model generates images of the desired text and style, and the performance is quite satisfying with recognizable characters and similar styles.

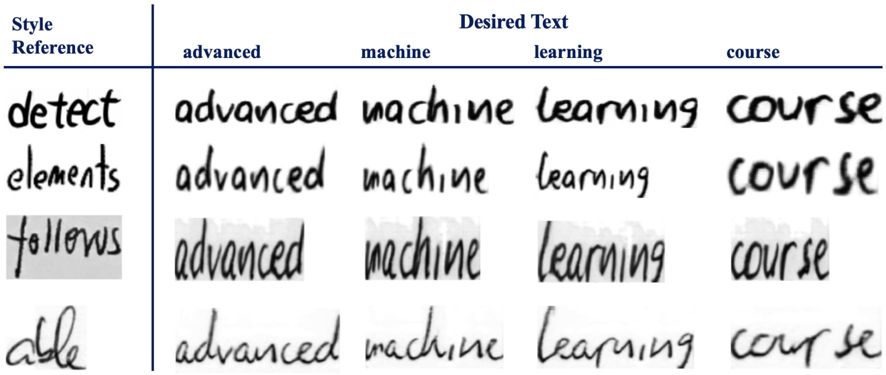

Figure 7: Representative generated results

6 Conclusion

In this study, our OCR-based pipeline enriched the handwriting dataset with tens of thousands of annotated images. Additionally, the LSTM-based style encoder successfully enhanced the HiGAN+ model to generate realistic handwriting images with desired calligraphic styles. These achievements mark significant progress in addressing data scarcity and enhancing the adaptability of handwriting analysis systems, with broad applications in signature authentication, document verification, and forensic analysis.

References

[1] Marti ZU-V and Bunke Horst. The iam-database: An english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.

[2] Ji Gan, Weiqiang Wang, Jiaxu Leng, and Xinbo Gao. Higan+: Handwriting imitation gan with disentangled representations. *ACM Trans. Graph.*, 42(1), 2022.

[3] Praveen Krishnan, Rama Kovvuri, Guan Pang, Boris Vassilev, and Tal Hassner. Textstylebrush: Transfer of text aesthetics from a single example, 2021.

[4] Lei Kang, Pau Riba, Yaxing Wang, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Ganwriting: Content-conditioned generation of styled handwritten word images, 2020.