Handwriting Generation and Animation with Deep Learning

Anita Dash
MA20BTECH11001

Dhruv Srikanth
EE20BTECH11014

Shambhu Prasad Kavir
CS20BTECH11045

Taha Adeel Mohammed
CS20BTECH11052

Abstract

Handwriting style transfer for personalized text generation is a fascinating and evolving field that sits at the intersection of computer vision and natural language processing. It has several applications in various domains, ranging from enhancing digital communication and marketing to improving accessibility and education.

In this project, we aim to create a system that takes a written text prompt and a sample of the handwriting to mimic as the input, and generates realistic animated handwritten text that closely resembles the provided style while maintaining readability and coherence.

1. Introduction

Handwriting is a deeply personal and authentic form of expression. It holds a unique place in human communication and self-expression. In this regard, Handwriting Style Transfer can come in handy. A model that generates handwritten text in a particular style can be helpful in the following fields:-

- In education, personalized handwritten materials can improve student engagement and retention, offering educators a powerful tool to make learning more effective and enjoyable.
- Handwriting style transfer can improve accessibility for individuals with motor disabilities, allowing them to convey their thoughts and emotions through personalized handwriting styles, promoting inclusivity in communication.
- Handwritten image generation can also provide additional data to train more accurate general handwriting recognition models
- It can help in the field of forensic analysis such as forgery detection, document authentication etc.
- Businesses can leverage personalized handwriting in marketing materials, enhancing customer engagement, and creating memorable brand experiences.

2. Problem Statement

Implement a deep learning model that can take a text prompt and a sample image of a person's handwriting style as input, and generate the image of the text prompt, in the specified handwriting style as the output. We want the model to accurately mimic the unique characteristics, nuances, and idiosyncrasies of the provided handwriting style while maintaining the readability and coherence of the generated text. We could consider further diversification - calculating the temporal sequence of brush strokes: if our generation model is successful, we could try to calculate the pen trajectory, i.e. the successive coordinate points for every time step of the brush stroke, to be extracted. This trajectory can then be used to animate the brush stroke in real-time, creating a realistic animated handwritten text.

3. Literature Review

There has been a lot of research in the field of handwriting generation and handwriting trajectory recovery. In this section, we discuss the various models and approaches that have been proposed in the literature to solve these problems. We then aim to combine these state-of-the-art techniques to create a model that can generate realistic animated handwritten text in a particular style.

3.1. Decoupled Style Descriptors

Capturing a space of handwriting stroke styles poses the challenge of representing both the style of each character and the overall style of the human writer. This paper [7] introduces an approach to online handwriting stroke representation via the Decoupled Style Descriptor (DSD) Model.

As handwriting strokes can be modeled as a sequence of points over time, supervised deep learning methods to hand-

writing representation can use recurrent neural networks (RNN).

In this approach, there are three variations represented within an RNN model: the variation in writer style, the variation in character style, and the variation in writer-character style. Given a database of timestamped sequences of handwriting strokes with character labels, this model learns a representation that encodes three critical factors:- writer-independent character descriptors, writer-dependent character string style descriptors and writer-dependent global style descriptors.

We have as input $x = (p_1, p_2, ...p_N)$ which represents the stroke sequence and $s = (c_1, c_2, ..., c_M)$ represents the character sequence. An unsupervised learning technique is used to train a segmentation network $k_\theta(x, s)$ to map regions in x to characters. We wish to predict x' comprised of p'_t. Further, Mixed Density Networks are used to provide variation in the output while generating the writing.

For the given x, s and a target string c_t a parameterized encoder function f_θ^{enc} is trained to learn writer-dependent character-dependent latent vectors w_{c_t}. Simultaneously, a parameterized decoder function f_θ^{dec} is trained to predict the next point p'_t given all the past points $p'_{1:t-1}$. Both the encoder and decoder functions used here are RNNs. To this method, so as to factor in character-independent writer style, we add another layer of abstraction, and introduce a parameterized encoder function g_θ.

Given a target character c_t, we use encoder g_θ to generate a C matrix. We then multiply C_{c_t} by a desired writer style w to generate w_{c_t}. Finally, we use a trained decoder f_θ^{dec} to create a new point p'_t given previous points $p'_{1:t-1}$.

$$p'_t = f_\theta^{dec}(p'_{1:t-1}|w_{c_t}), \text{where } w_{c_t} = C_{c_t} w$$

In a qualitative user study, it was observed that the drawing samples generated by this model were preferred over few of the state of the art techniques. Further the model was also successful in interpolating samples at different levels, recovering representations for new characters and achieved high-writer identification accuracy. Despite that, the model occasionally failed in producing legible letters or in connecting cursive letters. One of the causes for this issue being the underlying inconsistencies in human writing, which was only partially addressed in this model. Additionally, the process of collecting high-quality data using digital pens in a crowdsourced environment, involving careful data cleaning, persists to be another challenge.

3.2. GANwriting

Generative Adversarial Networks (GANs) have been succesfully used for generating illusory plausible images in various fields. GANs consist of two neural networks, a generator, and a discriminator, which are trained simultaneously through a competitive process in which both improve iteratively.

In the paper [6], the authors use a conditional non-recurrent generative adversial (cGAN), to produce realistic handwritten word images. In order to produce these diverse stylized words, the textual content along with the specific wrting style, defined by a latent set of calligraphic attributes, are separately conditioned on the generative model. To train the model and achieve the desired results, the authors used the following three novel techniques:

- Three complementary learning objectives, namely adversial loss, style classification loss, and reconstruction loss, are used to train the model. State-of-the-art discriminator, classifier, and word recognizer networks are used to train the model.
- Character-based content conditioning is done, allowing to generate any word, without being restricted to a specific vocabulary.
- Few-shot calligraphic style conditioning is done to avoide the mode collapse problem.

However this model is limited to singular words. In the paper [4], they expand on these ideas to allow variable length textual input, allowing it to generates entire lines of offline handwriting.

3.3. Handwriting Transformers

Earlier Handwriting generative methods process style and features separately. It doesn't encode style content entanglement at a character level. In this paper [3]. the authors propose a transformer-based styled handwritten text image generation approach, HWT, that strives to learn both style-content entanglement as well as global (such as ink width, slant) and local (such as character style, ligatures) writing style patterns. The overall architecture has four components:

- Conditional Generator : Synthesize handwritten text.
- Discriminator: Ensures realistic generation of handwriting styles. It is designed to be convolutional in nature
- Recognizer: Aids in textual content preservation. It is inspired by CRNN
- Style Classifier: Ensures satisfactory transfer of calligraphic styles.

In the paper, the focus of the design is in the generator model. To imitate a handwriting style as realistically as possible, This model is designed to learn style content entanglement as well as local and global style patterns. It is

a transformer-based generative network for unconstrained styled handwritten text image generation. It has two main components an encoder network and a decoder network. Both the encoder and decoder networks constitute a hybrid design based on convolution and multi-head self-attention networks.

HWT generates realistic styled handwritten text images and significantly outperforms other state-of-the-art models through extensive qualitative, quantitative and human based evaluations. The model also generalizes well to the challenging scenarios where both words and writing style are unseen during training, generating realistic styled handwritten text images

3.4. Dynamically configurable CRNN (Recognizer)

The DC-CRNN (Dynamically Configurable Convolutional Recurrent Neural Network) [1] is one of the best performing handwriting recognition models out now. The CRNN has been used as a recognizer in other models as well. Recognition is a vital prerequisite to generation so we can learn how to better generate handwritten text from better recognition models.

The CRNN structure counters the most significant challenge with handwriting analysis which is variable-length sequences. A CNN module is used to extract spatial features while the label, text data corresponding to the image, is treated as a character sequence fed to the RNN. With this setup, the model is able to accurately learn correlation between different characters while considering the global effect of the handwriting style via its CNN module. The optimization problem is framed differently using SSA and LAHC to have a more generalized solution that considers more of the data than exploit it using swarm optimization given the complexity of the latent space. Though the algorithm itself requires more investigation to understand its specific value, the result is shown to significantly improve when LAHC (Late Acceptance Hill Climbing) is used - comparing current solution with that from multiple steps ago to ensure stability.

Other models are limited by the sequence and this architecture appears to solve that long-standing issue. We want to explore possible integration of these techniques in generation models for improved performance.

3.5. Handwriting Trajectory Recovery

Temporal information is unavailable when it comes to offline text. An image scanner or a video camera is not capable of extracting information like velocity, pressure, inclination etc. If one is able to recover the stroke trajectory from the static 2D image, then offline text can be viewed as an online text. This paper [2] proposes a technique that can predict the probable trajectory of an offline character level image.

The model is inspired from the sequence to sequence model based on encoder-decoder architecture. The model sequentially predicts the data coordinate points of the pen trajectory from the offline character images.The framework of this model consists of mainly two steps.

- Extract a sequence feature vector from the offline images using CNN.
- An encoder-decoder LSTM network takes the sequence feature as the input and outputs the required coordinate points.

The model is able to find out the correct starting point and extract the correct incoming and outgoing paths from junction points effectively.

3.6. Domain-Adversarial Neural Network

DANN (Domain-Adversarial Neural Network) [5] is an interesting architecture that we believe, could be utilized in the problem of specific-style handwriting generation. DANN is an improvement to any regular network structure due to its parallel branch.

Given a Neural Network performing a downstream task on a given dataset, the output of the penultimate layer is considered the final representation of the input data. On this representation, we apply our last layer which is the downstream task itself but the more interesting part is the learnt representation. We claim that with enough training, this penultimate layer output is the best possible latent space representation of the data distribution possessing the information required to perform a downstream task. The nature of Deep Learning makes it so that we can never truly know the kind of correlations the model has learnt from the input data and how the information is being represented but we know that the information required for the downstream task is somehow contained in the representation. If the existence of a downstream task and its loss function controls the information present in the representation, then if we want to include or exclude further information from the representation, all we need to do is have a parallel branch of the neural network performing that downstream task on the representation. When we consider the loss of this new branch downstream task while updating our representation-learning, the learnt representation now possesses or is independent of information regarding this task.

For example, if we had images of X-Rays of a human body part and we had to determine if the subject is fractured or not, our primary downstream task would be to classify the image as fractured or not. But if we have labels as to what part of the human body is present in the image, then we can ensure that our representation learns features of the image that are independent of whether the subject is a hand or a leg. This can be implemented by having a parallel

downstream task on the representation that classifies the image as being an X-Ray of a hand or an X-Ray of a leg. Next, we subtract the loss of this classifier for the representation-learning layers. Thus, the model is forced to get worse at identifying whether the subject in the image is a hand or a leg making its learnt representation independent (but aware) of this information and only focussed on whether the subject is fractured or not.

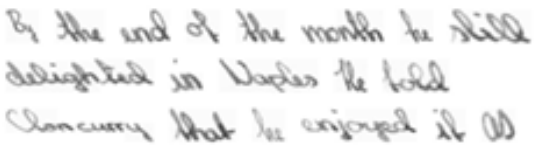
Thus, this architecture allows us to hand-pick the kind of information we want to be depicted in the model's latent space representation of the data by making the model either improve or become worse at a parallel task involving correlated information. This architecture was originally proposed for use in the biomedical space but we believe that style transfer problems too could benefit from such an architecture due to the requirement of considering handwriting style information and character information distinctly. If we are able to isolate the style information from the character information completely giving the model a better empirical understanding of the handwriting style itself, this may improve our ability to generate better handwriting samples.

3.7. Stroke Paper

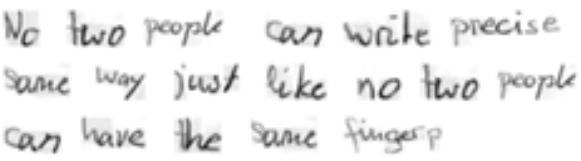
Add Info

4. Replicated Results

GANwriting [6] is limited to generating singular words, and as a result we cannot have outputs of variable length. This is a major drawback as we want to generate entire lines of text. It doesnt encode style content entanglement at a character level and hence struggles to mimic character specific styles.



(a) **Desired Handwriting**

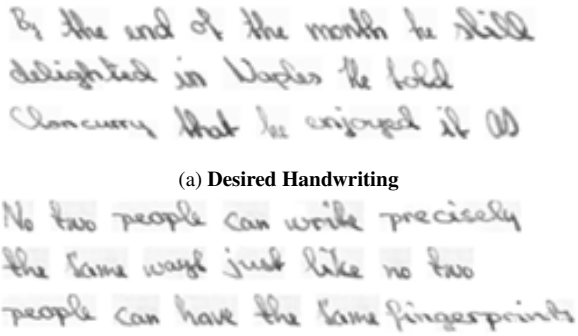


(b) **GANwriting**

Figure 1. **Mention Interpretation**

While Decoupled Style Descriptors [7] is able to capture local and global style patterns, it fails to generate legible letters or connect cursive letters. This is because of the underlying inconsistencies in human writing, which is only partially addressed in this model. Additionally, the model works with online handwriting data and we want to capture the styles of offline handwriting.

Handwriting Transformers [3] is able to generate realistic styled handwritten text images and significantly outperforms other state-of-the-art models. It handles variable length input and captures local and global style patterns. It is also compatible with offline handwriting data. Hence we have decided to use this model as our base model.



(a) **Desired Handwriting**

(b) **HWT**

Figure 2. **Mention Interpretation**

4.1. FID Scores

Explanation of FID scores Add results

4.2. Testing

Add results

5. Preliminary Results

Testing with our own handwriting
Add changes made to make testing more accessibile

References

[1] Ahmed AL-Saffar, Suryanti Awang, Wafaa AL-Saiagh, Ahmed Salih AL-Khaleefa, and Saad Adnan Abed. A sequential handwriting recognition model based on a dynamically configurable crnn. *Sensors*, 21(21), 2021. 3

[2] Ayan Kumar Bhunia, Abir Bhowmick, Ankan Kumar Bhunia, Aishik Konwer, Prithaj Banerjee, Partha Pratim Roy, and Umapada Pal. Handwriting trajectory recovery using end-to-end deep encoder-decoder network. pages 3639–3644, 2018. 3

[3] Ankan Kumar Bhunia, Salman H. Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Mubarak Shah. Handwriting transformers. *CoRR*, abs/2104.03964, 2021. 2, 4

[4] Brian L. Davis, Chris Tensmeyer, Brian L. Price, Curtis Wigington, Bryan S. Morse, and Rajiv Jain. Text and style conditioned GAN for generation of offline handwriting lines. *CoRR*, abs/2009.00678, 2020. 2

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. 2016. 3

[6] Lei Kang, Pau Riba, Yaxing Wang, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Ganwriting: Content-conditioned generation of styled handwritten word images. 2020. 2, 4

[7] Atsunobu Kotani, Stefanie Tellex, and James Tompkin. Generating handwriting via decouple style descriptors. *CoRR*, abs/2008.11354, 2020. 1, 4