Handwriting Generation and Animation with Deep Learning -Midterm Report-

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. It is also an interesting and challenging problem to work with for DL models. There has been extensive research in this field, both very recently, and in the past. There exist various state-of-the-art models solving the problems of handwriting recognition, handwrting classification, handwriting generation, handwriting style transfer, and handwriting trajectory recovery. In this project, we aim to combine these state-of-the-art techniques to create a model that can generate realistic animated handwritten text in a particular style.

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 in the given style.

3. Literature Review

Preliminary Report Literature Review

3.1. Decoupled Style Descriptors [9]

3.2. GANwriting [4, 8]

3.3. Handwriting Transformers [3]

3.4. Dynamic CRNN (Recognizer) [1]

3.5. Handwriting Trajectory Recovery [2]

3.6. Domain-Adversarial Neural Network [6]

,

Midterm Report Literature Review

3.7. Writing Order Recovery [5]

Writing order recovery is a complex problem that relates to the intrinsic properties of human handwriting. This paper [5] proposes an innovative deterministic algorithm to recover the writing order of any thinned long static handwritten signature. Signatures have been chosen out of the belief that they are the most complex version of this problem.

The proposed method is completely intuitive and draws from the good continuity criteria of handwriting. The process of recovery has been split into 3 subprocesses - point classification, local examination, and global reconstruction.

Strokes are believed to be entirely continuous so the authors have observed the 8-connected pixels, adjacent pixels surrounding a target pixel, to observe the target pixel's position within a stroke. A pixel with 2 connected pixels is believed to be a trace-point, one of the points along the trajectory of a stroke. A point with only 1 connected pixel is believed to be an end-point of a stroke. Any point with more than 2 connected pixels is believed to be a cluster point, a point found in the region of overlap between 2 distinct strokes/components. After this classification, the next step is described as local examination. Adjacent trace points are considered to be part of one stroke and hence form large groups reconstructing the strokes from an end-point, through trace-points, to another end-point. The only complexity remaining at this stage is that of overlapping strokes forming clusters. Branches exiting clusters are marked with anchor points and their exit angles are measured and characterized. The authors define a few commonly occurring cluster scenarios and match the present observed cluster to the predefined scenarios. In the global reconstruction stage, clusters are resolved by modelling the rapid change in direction (from branch angle and position) as energy alongside assigning priority to each branch and each scenario to perform energy minimization. After the internal cluster paths are separated, the authors use a Gaussian spread formulation to choose the leftmost starting point of a stroke and use a proximity criterion to connect a pen-up point to the next pen-down point thus deriving the order of strokes/components.

The main complexity of such a problem has been described as the interpolation of pen-up and pen-down amidst strokes causing distinct strokes and consequently their overlap. The solution proposed to this problem is intuitive and replicative of a human thought process which is why it captured our attention.

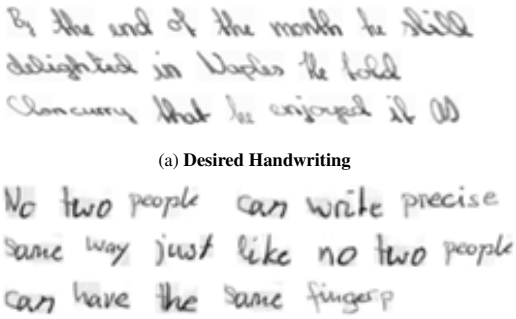
4. Our approach

Our final desired output is a series of points describing the trajectory of the generated handwriting. For this, cRNN models such as [1, 9], which predict the next point in a sequence, given the earlier points, directly produce our desired output. However, recently, there have been better state-of-the-art models for handwriting generation, such as [3], which use transformers. Hence we use the HWT model [3] as our base model to generate the image of the text prompt in the specified handwriting style. We then use a handwriting trajectory recovery model [2, 7] to recover the trajectory of the generated handwriting. Hence we are able to get more realistic animated handwritten text in the specified handwriting style, as opposed to directly using the output of cRNN models.

5. Replicated Results

We have replicated the results of the HWT model [3], Decoupled Style Descriptors [9], and GANwriting [8] models based on the papers and the code provided by the authors. Below we compare these three models.

GAN Writing



(a) **Desired Handwriting**

(b) **GANwriting**

Figure 1. Since GANwriting is limited to generating fixed number of words, it fails to generate the entire line of text. It also fails to capture the style of the text.

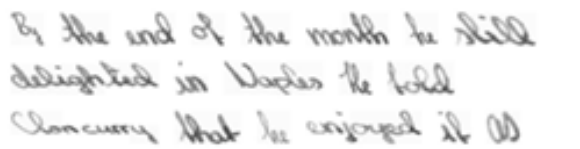
GANwriting [8] is limited to generating singular words, and as a result we cannot have outputs of variable length. This is a major drawback as we want to generate entire lines of text. It doesn't encode style content entanglement at a character level and hence struggles to mimic character specific styles. Fig.1 tests the model on the given desired style.

Decoupled Style Descriptors

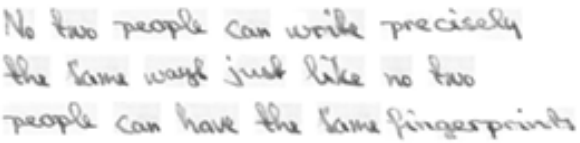
According to [9] Decoupled Style Descriptors is able to capture local and global style patterns, it fails to generate legible letters or connect cursive letters. This is because of the underlying inconsistencies in human writing, which is only partially addressed in this model. Additionally, the model works with online handwriting data and we want to capture the styles of offline handwriting.

Handwriting Transformers

The Handwriting Transformers model [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. Fig.2 tests the model on the given desired style.



(a) **Desired Handwriting**



(b) **HWT**

Figure 2. HWT is able to generate the entire line of text. It also captures the local and global style of the text.

FID Scores

The Fréchet Inception Distance (FID) is a metric used to evaluate the quality of generated images. The FID score is the Frechet Distance between the distribution of the real images and the distribution of the generated images. The lower the FID score, the better the quality of the generated images. A score close to zero implies that the two groups of images are nearly identical.

The IAM dataset [10] is used to compute the FID scores. The IAM dataset contains 1,539 pages of scanned text from 657 writers. We compute the score between the generated images and the real images from the IAM dataset.

Table 1: Comparison of HWT [3] with GANwriting [8] model with respect to their FID scores computed between the generated text images and the real text images from the IAM dataset. We generate datasets of 6 images each for both the models. The FID score for the HWT model is lower than that of the GANwriting model implying that the former generates better quality images than the latter.

Model	FID Score
HWT	70.67
GANwriting	94.77

While the size of the datasets used here was rather small, we can still observe that the HWT model generates better quality images than the GANwriting model. This is because the HWT model is able to capture the local and global style patterns of the text, while the GANwriting model is not. The purpose of this test was only to attest what was already shown in the papers. The code for the above test can be found here.

Table 2: Comparison of HWT model with ScrabbleGAN and Davis et al with respect to their FID scores. We generate a dataset of 10,000 images for the HWT model and then compute its FID score. For the other models, we take the FID scores as mentioned in the paper [3]. Once again, we observe that the HWT model has a lower FID score than the other models.

Model	FID Score
HWT	16.71
ScrabbleGAN	20.72
Davis et al	20.65

We again observe that the HWT model has a lower FID score than the other models. The code for the above test can be found here.

6. Proposed Model

6.1. Architecture Overview

The model architecture described in the Handwriting Transformers paper is as follows. The sample image of the required handwriting is fed to the transformer encoder. The encoder consists of a ResNet-18-based CNN encoder network. The encoded feature set obtained is then passed to a transformer network of embedding size $d = 512$ with 3 attention layers having 8 attention heads thus concluding the handwriting distillation module. The final transformer-encoded feature vector is passed to the transformer decoder to begin handwriting generation.

The query prompt is encoded to form a query embedding and passed to the transformer decoder along with the handwriting-style feature vector. The transformer decoder of similar attention structure as the transformer encoder outputs a decoded feature vector which is passed to a ResNet-18-based CNN decoder to generate a full-size image. This generated image is passed to a discriminator, a CRNN-based recognizer, and an ANN-based style classifier - all in place to fact-check the generator.

6.2. Loss Function

The loss function of this network, though complex, when broken down and related to each component of the network, is fairly intuitive. The model can be thought of as two distinct processes - extracting the required handwriting from a sample and applying the distilled handwriting to a new text prompt. The handwriting distillation module is constructed using a cycle loss. The code obtained from the encoder is used to reconstruct the image and clarify that all important details of the image can be reconstructed from this code determining that the code possesses all the required information about the handwriting.

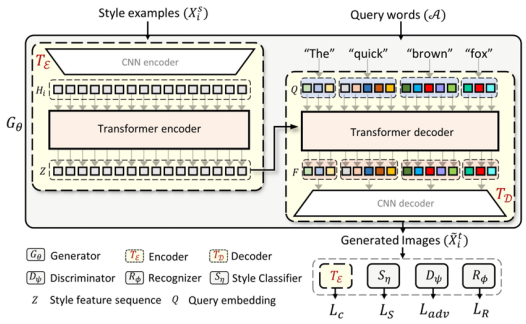


Figure 3. Handwriting Transformer model architecture

The loss function of the handwriting generation module has three main components. First, the generative adversarial loss is defined by the discriminator's ability to distinguish a generated image coming out of the decoder from a dataset-fed image thus ensuring that any newly generated image bears a likeness to the dataset. Therefore, the first loss component relates to the discriminator.

The second loss component is defined by the recognizer. The recognizer is tasked with ensuring the legibility of the generated handwritten text in the sense that individual letters in the generated handwritten image are clearly distinguishable and recognizable. The text of the generated image, the ground truth used by the recognizer, is supplied as a query text prompt to the generator model. Treating this text prompt as a sequence of characters, the recognizer uses CTC loss, connectionist temporal classification loss (a sequentially modeled classification loss), to ensure that each character in the sequence is correctly manifested. In a sense, it consists of a classifier that attempts to classify each character in the generated image as one of the known characters of the English language ensuring that the language details are retained.

The third loss component is a cross-entropy classification loss defined by the style classifier. The classifier attempts to classify the generated image into one of its known handwriting styles thus ensuring that unique handwriting details corresponding to the specific author of the original sample are retained within the image.

7. Testing on IAM-Dataset

We test the Handwriting Transformers model on the IAM dataset [10]. We feed the IAM dataset as input to the model with pretrained weights, and generate images in the handwriting styles of the writers in the IAM dataset, on which the model was originally trained.

We use a batch size of 8, i.e. we get 8 writer styles per page. The prompt used was *Magic madness heaven sin Saw you there and I thought*. We can observe that the model is able to generate quite realistic images, capturing both global and local style patterns. The code for generating images, and more such examples, can be found here.

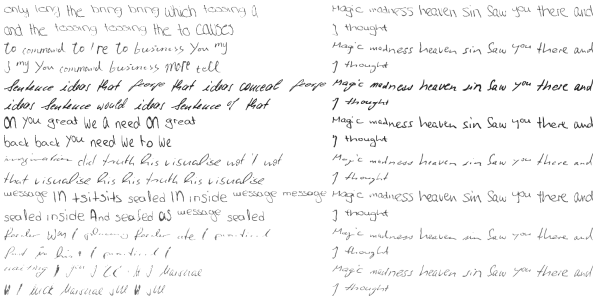


Figure 4. Example-1

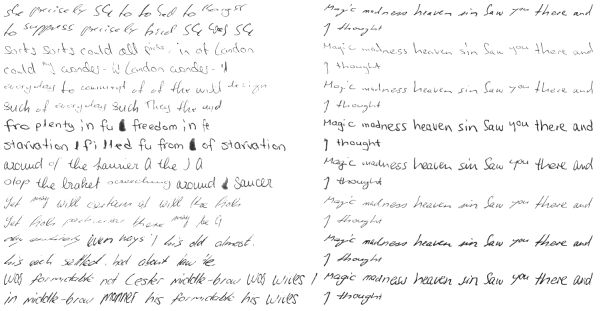


Figure 5. Example-2

8. Preliminary Results

The HWT model used to replicate the results in the previous section is trained on the IAM dataset [10] and only accepts segmented word images from the target writer as the style input. We build on this model's testing to accept a text prompt and a sample of the handwriting to mimic from any writer, no longer constrained to belong to the IAM dataset. This demo code to perform custom testing can be viewed here.

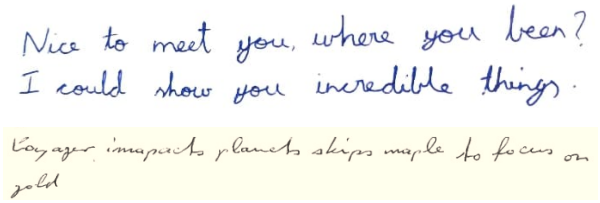
Furthermore, we developed `para_to_word.ipynb`, which extracts the segmented word bounding boxes from a given paragraph image. This is done by using the OpenCV library, and using its functionality to find the contours in the image. We then cluster the nearby contours together to form the word bounding boxes. These word bounding boxes can then be fed into the HWT model to generate the desired output. Hence we are able to use a paragraph image as the input to the HWT model, instead of a single word images. Below we show a sample word extraction from a paragraph image.

Using above functionality, we run the model on handwritings of the authors of this report, and successfully mimic their style on our text promt, as shown in 7

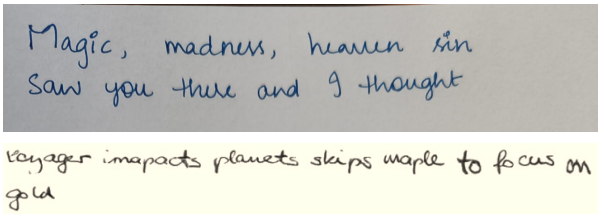
(a) Input Image

(b) Words

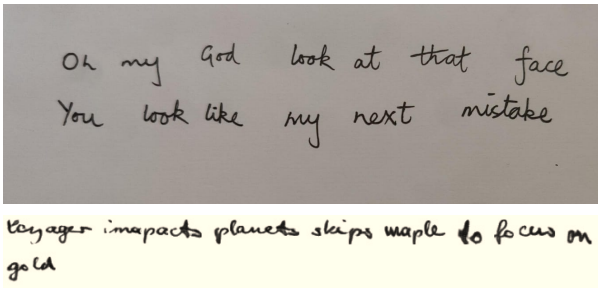
Figure 6. Using Word Bounding Boxes to split into segmented word images



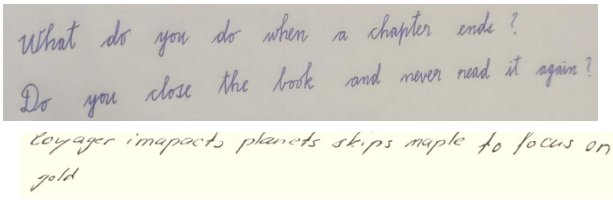
(a) Writer: Taha



(b) Writer: Anita



(c) Writer: Dhruv



(d) Writer: Shambu

Figure 7. We tested the model's performance using our own handwriting. Text prompt used: "Voyager impacts planets skips maples to focus on gold".

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.

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

[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.

[5] Moises Diaz, Gioele Crispo, Antonio Parziale, Angelo Marcelli, and Miguel A. Ferrer. Writing order recovery in complex and long static handwriting. *International Journal of Interactive Multimedia and Artificial Intelligence*, X(X):1–14, 2021.

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

[7] Sidra Hanif and Longin Jan Latecki. Strokes trajectory recovery for unconstrained handwritten documents with automatic evaluation. pages 661–671, 01 2023.

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

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

[10] Urs-Viktor Marti and H. Bunke. The iam-database: An english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39–46, 11 2002.