A COMPREHENSIVE FRAMEWORK FOR STROKE TRAJECTORY RECOVERY FOR UNCONSTRAINED HANDWRITTEN DOCUMENTS

A Dissertation
Submitted to
the Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY

by
Sidra Hanif
May 2024

Examining committee members:
Dr. Longin Jan Latecki, Advisory chair, Department of Computer and Information Science
Dr. Richard Souvenir, Department of Computer and Information Science
Dr. Stephen MacNeil, Department of Computer and Information Science
Dr. Haibin Ling, Department of Computer and Information Science
Dr. Iyad Obeid, External Member, Department of Electrical and Computer Engineering

Abstract

For a long time, handwriting analysis, such as handwriting recognition and signature verification, has been an active research area. There are two categories of handwriting, online and offline. Online handwriting is captured in real-time on a digital device such as a tablet screen with a stylus pen. In contrast, the handwritten text scanned or captured by a camera from a physical medium such as paper is referred to as offline handwriting. For offline handwriting, the input is limited to handwritten images, making handwriting analysis much more difficult. In our work, we proposed a Stroke Trajectory Recover (STR) for offline and unconstrained handwritten documents. For this purpose, we introduce large-scale word-level annotations for the English handwriting sampled from the IAM-online dataset. The current STR architectures for English handwriting use lines of text or characters of the alphabet as input. However, a word-level STR method estimates loss for each word rather than averaging DTW loss over the entire line of text. Furthermore, to avoid the stray points/artifacts in predicted stroke points, we employ a marginal Chamfer distance that penalizes large, easily noticeable deviations and artifacts. For word detection, we propose the fusion of character region scores with bounding box estimation. Since the character level annotations are not available for handwritten text, we estimate the character region scores in a weakly supervised manner. Character region scores are estimated autonomously from the word's bounding box estimation to learn the character level information in handwriting. We propose to fuse the character region scores and images to detect words in camera-captured handwriting images. We also propose an automated evaluation to check the quality of the predicted stroke trajectory. The existing handwriting datasets have limited availability of stroke coordinates information. Hence, although the proposed system can be applied to handwriting datasets without stroke coordinates information, it is impossible to evaluate the quality of its predicted strokes using the existing methods. Therefore, in our work, we propose two measures for evaluating the quality of recovered stroke trajectories when ground truth stroke information is not given. First, we formulated an automated evaluation measure based on image matching by finding the difference between original and rendered images. We also evaluated the preservation of readability of words for original and rendered images with a transformer-based word recognition network. Since our proposed STR system works with words, we demonstrate that our method is scalable to unconstrained handwritten documents, i.e., full-page text.

Finally, we present a probabilistic diffusion model conditioned on handwriting style template for generating writing strokes. In handwriting stroke generation, imitating a calligraphic style of template image has significant importance. However, previous studies have not emphasized the calligraphic features of the handwriting style, which in turn results in inadequate style imitation ability of the learning model. In our work, we propose to utilize a strong multi-scale feature for calligraphic style extraction. We also introduce a character

and character-pair style features to include local and global style features for handwriting stroke generation. Conventional handwriting image evaluation methods are based on evaluations designed on natural images, which do not capture global writing style and character shape. In our work, we propose style features and projected character shape matching for the evaluation of handwriting stroke generation.

Moreover, we train our diffusion model for handwriting stroke prediction with the Dynamic Time Warping (DTW) loss function, along with the diffusion loss, which eliminates the need to train any auxiliary networks for text or writer style recognition and adversarial network. Our experimentation shows that the proposed conditional diffusion model trained with multiscale attention style features and dynamic time Warping outperforms the current state-of-the-art stroke generation network.

I would like to dedicate my work to all my teachers and my parents.

Contents

Content **I**

List of Figures **III**

List of Tables **IV**

Introduction **1**

1 Literature Review **5**

1.1 Word detection 5

1.1.1 Scene text detection 5

1.1.2 Object detection 6

1.2 Stroke trajectory recovery 6

1.3 Handwriting image generation from style image 8

1.4 Handwriting strokes generation from style image 9

1.5 Contribution 10

2 Word Detection in Camera-captured Handwriting Documents **11**

2.1 Method 11

2.1.1 Multi-channel word detector 12

2.2 Experiments 14

2.2.1 Datasets 14

2.2.2 Results and discussion 14

2.3 Conclusion 17

3 Strokes Trajectory Recovery for Unconstrained Handwritten Documents **18**

3.1 Method 18

3.1.1 Network architecture 18

3.1.2 Loss function 18

3.2 Experiments 20

3.2.1 Word-level annotation 21

3.2.2 Evaluation metric 22
3.3 Results 24
3.3.1 IAM-online dataset 24
3.3.2 Greeting cards messages dataset 25
3.4 Conclusions 27

4 Automatic evaluation 28
4.1 Image matching 28
4.2 Readability 29
4.3 Experiments 30

5 Local Feature Fusion with Character Shape Refinement for Handwriting Imitation 32
5.0.1 Problem statement 34
5.1 Method 35
5.1.1 Network architecture 35
5.1.2 Loss functions 37
5.2 Experimentation 38
5.2.1 Dataset 38
5.2.2 Results 38

6 MS-CAP: Probabilistic Diffusion Model Conditioned on Multi-Scale ChAracter Pairs for Style in Handwriting Stroke's Generation 40
6.1 Method 40
6.1.1 Multi-scale attention features 41
6.1.2 MSCAP text-style encoder 45
6.1.3 Stroke diffusion Model 47
6.1.4 Loss function 49
6.2 Experiments 51
6.2.1 Strokes-based metrics 51
6.2.2 Image-based metrics 52
6.2.3 Results 55
6.2.4 Discussion 58
6.3 Conclusion 59

Bibliography 59

List of Figures

1 The input textual content and calligraphic style for handwriting stroke generation are fed into the proposed system. The output stroke of the system imitates the input textual content in the given calligraphic style. 3

2.1 (a) character region + affinity score (b)word detection based on (a) 12

2.2 The block diagram of the convolution network comprising encoder, decoder and detection branches. Encoder-decoder pair is used to learn character region scores and detection branch learn the multi-channel word detection for fused character region scores and handwriting image. The break symbol between encoder-decoder branch and detection branch shows the autonomy of both branches. 13

2.3 Sample images from GNHK datasets [49] with bounding boxes 16

2.4 The qualitative results for word detection for low-contrast camera -captured handwritten text 16

3.1 The block diagram of the overall system for training, inference, and automatic evaluation. 19

3.2 The impact of marginal Chamfer distance loss on the stroke trajectory recovery. (a) DTW loss only. (b) DTW + Chamfer distance loss. 20

3.3 (a) Word detection for the GNHK dataset [49], (b) word detection for IAM-online dataset [54], (c) Words from IAM-online dataset used to train our network. 22

3.4 Sample documents form GHM dataset with unlabelled handwriting images. . 23

3.5 Samples of stroke trajectory recovery with (a) DTW loss and (b) DTW loss with Chamfer distance on the predicted and ground truth point sets. The recovered stroke trajectory is shown by red, blue, and green arrows, and the predicted start-of-stroke point is shown as an orange circle (best view in colored). 24

3.6 The mechanism of image rendering from predicted stroke points. 25

3.7 The input image and the rendered image from predicted stroke trajectory recovery for GHM dataset. 26

4.1 The readability module based on text recognition [50] utilized for our automatic evaluation. The top-left text is the input and the bottom-left text is reconstructed from predicted stroke trajectory recovery. 30
4.2 The input image, dilated image (after dilation is applied to the recovered image), and the difference of the input and dilated images. 31

5.1 The overall network architecture 33
5.2 The generated and ground-truth image and their respective histograms . . . 34
5.3 The quantitative comparison of previous [26] and proposed method 39

6.1 The overall block diagram of diffusion model with multiscale attention features for the style image. It also includes Dynamic Time Warping (DTW) loss for stroke sequence matching to train the diffusion model.) 40
6.2 The architecture for multi-scale learning with attention via writer identification. 42
6.3 (a) Block diagram of a text-style encoder for stroke diffusion [53], (b) Block diagram of a text-style encoder for modifying style features with character style as well as the connection between characters. 45
6.4 The internal layer-wise architecture of diffusion model for strokes prediction from handwriting images for training and inference phase. 46
6.5 A real image and a generated image, and their respective histograms. We show all the six rotated image pairs and their histogram. 50
6.6 Visual comparison with previous image generation methods. 54
6.7 Visual comparison of the proposed method with the state-of-the-art handwriting image generation methods for offline handwriting word samples. 55
6.8 Visual comparison of multi-scale attention style features and the Dynamic Time Warping (DTW) loss. 57
6.9 Exemplar qualitative results of image generation through stroke generation on the IAM-online dataset. For each example, the text content is shown in a dotted rectangular block with the input style image and generated images below it. 60

List of Tables

1 The capabilities of the previous and proposed methods for handwriting image and stroke generation 2

2.1 The detection accuracy for two-stage object detector [67] and Character region score [7] 14
2.2 The detection statistics for image, character region and affinity score on multichannel object detector network [12]. In the table, image stands for 3-channel handwriting image, RS stands for character region score and AS stands for character affinity scores. 15
2.3 The detection statistics for image and character region scores as input for the object detector network [12] for large, medium, and small handwriting sizes . 15

3.1 The accuracy (mAP) of word detection with state-of-the-art scene text detector [7] and single and two-stage object detectors [67, 12]. 21
3.2 The quantitative comparison for training on DTW loss for lines and words. The third row lists the quantitative results for training with combined DTW and Chamfer distance loss between predicted and ground-truth points 23
3.3 The quantitative comparison of increasing the value of c from 1 to 3. 23

4.1 The quantitative analysis of automatic evaluation on greeting cards handwritten messages [72]. 31

5.1 The quantitative comparison of our method with state of the art methods for the handwriting imitation 36
5.2 The ablation study for different loss functions for handwriting imitation . . . 36

6.1 Classification accuracy for writer identification for online and offline handwriting. The multi-scale attention outperforms the baseline for both online and offline by a large margin. 44
6.2 The distance of nearest predicted to GT points $dist_{p,t}$ and the distance of nearest GT to predicted points $dist_{t,p}$ for different style features. 51
6.3 Long sentences: Quantitative comparison of our method with state-of-the-art methods for handwriting imitation. The style input is an online handwritten image from the IAM-online dataset. 51
6.4 Words only: Quantitative comparison of our method with state-of-the-art methods for handwriting imitation. The style input is an offline handwritten image from the IAM-offline dataset. 51
6.5 Comparison of model size and architecture with the state-of-the-art methods. 57

Introduction

The focus of our work is offline handwriting Stroke Trajectory Recovery (STR), which facilitates the tasks such as handwriting recognition and synthesis. The input is an image of handwritten text, and the output is a stroke trajectory, where each stroke is a sequence of 2D point coordinates.

Recently, detecting and recognizing a handwritten text has gained much attention from the research community. However, word detection from unconstrained low-contrast camera-captured images is still an open problem in document analysis. Word detection from handwritten text plays a crucial role in the success of subsequent applications such as word recognition or reconstruction. Word detection is considered an object detection problem. However, characters are the basic building block in words, and the presence of characters makes word detection different from general object detection problems. Character region scores identification performs consistently for handwritten text in low-contrast camera-captured images, But detecting words from characters poses a challenge because of variable character spacing in words. Nevertheless, considering the only character and ignoring a word's entirety does not cope with overlapping words in handwriting text. In our work, we propose the fusion of character region scores with word detection. Since the character level annotations are not available for handwritten text, we estimate the character region scores in a weakly supervised manner. Character region scores are estimated autonomously from the word's bounding box estimation to learn the character level information in handwriting. Therefore, we propose to fuse the character region scores and images to detect words in camera-captured handwriting images.

Stroke Trajectory recovery (STR) is considered a sequence prediction problem where the input is a handwriting image and the output is the sequence of predicted stroke points. Usually, Dynamic Time Warping (DTW) or Euclidean distance-based loss function is employed to train the STR network. In DTW loss calculation, the predicted and ground-truth stroke sequences are aligned, and their differences are accumulated. The DTW loss penalizes the alignment of far-off points proportional to their distance. As a result, DTW loss incurs a small penalty if the predicted stroke sequence is aligned to the ground truth stroke sequence but includes stray points/ artifacts away from ground truth points. To address this issue,

Method	Image	Strokes	Random style	Desired style	Short words	Long sentences
Handwriting stroke prediction						
Base LSTM [11]		✓				
Trace [5]		✓	✓		✓	
U-STR [34]		✓	✓		✓	✓
Handwriting image generation						
HiGAN+ [26]	✓			✓	✓	✓
ScrabbleGAN [22]	✓		✓		✓	
VATr [59]	✓		✓		✓	
Wordstylist [57]	✓		✓		✓	
Handwriting stroke generation						
Brush [45]		✓				
Stroke diffusion [53]		✓			✓	
Ours		✓	✓	✓	✓	✓

Table 1: The capabilities of the previous and proposed methods for handwriting image and stroke generation

we propose to compute a marginal Chamfer distance between the predicted and the ground truth point sets to penalize the stray points more heavily. Our experiments show that the loss penalty incurred by complementing DTW with the marginal Chamfer distance gives better results for learning STR. We also propose an evaluation method for STR cases where ground truth stroke points are unavailable. We digitalize the predicted stroke points by rendering the stroke trajectory as an image and measuring the image similarity between the input handwriting image and the rendered digital image. We further evaluate the readability of recovered strokes. By employing an OCR system, we determine whether the input image and recovered strokes represent the same words.

The second section of our work focuses on handwriting stroke generation. While it is a difficult to generate an image from a given stroke sequence, the task of converting a handwriting image into a stroke sequence is very challenging since it facilitates many subsequent tasks, such as handwriting recognition [21] and writing order prediction [58]. In handwriting stroke generation, the desired calligraphic style is provided in the form of an image with a string of textual content. The system intends to learn to imitate the handwriting style for unseen textual content. Apart from mimicking the handwriting calligraphic style for stroke generation, the strokes are also required to generate readable text. Despite the advancements of image generation models [56, 64, 38] for natural scene generation, precisely representing the calligraphic style of handwriting images in a generation network is still an open problem. In a similar problem of handwriting stroke prediction, a network is designed to learn the stroke trajectory from handwritten images. When given only handwritten images, the network predicts the sequence of strokes drawn by the writer to write the given text. The ability of the previous handwriting stroke prediction networks [5, 34, 45] is limited to recovering the stroke trajectory from images. To estimate the stroke sequence of any arbitrary text in a desired style, we have to provide that text written in that particular calligraphic style. Since these networks are not conditioned on arbitrary text, their implementation depends

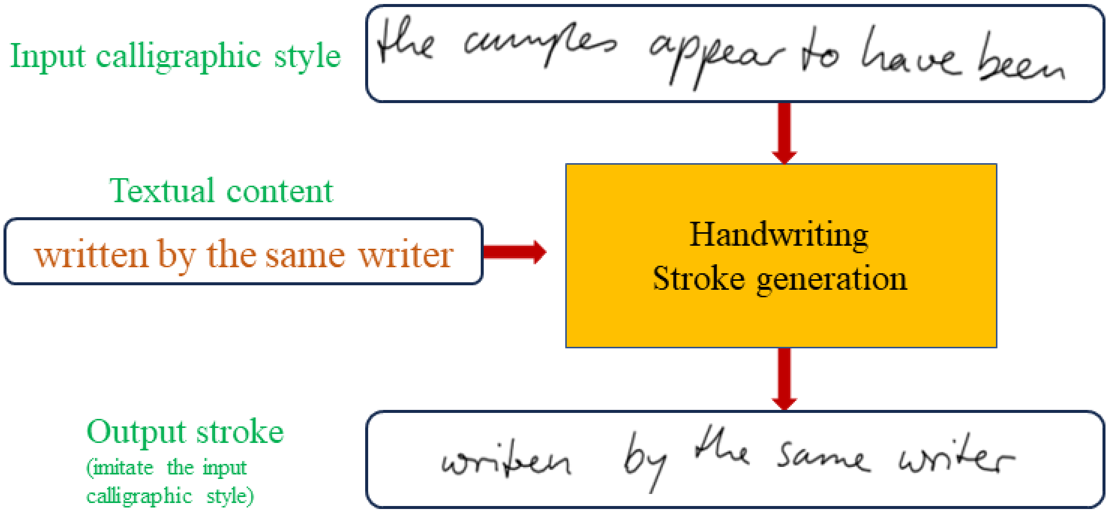


Figure 1: The input textual content and calligraphic style for handwriting stroke generation are fed into the proposed system. The output stroke of the system imitates the input textual content in the given calligraphic style.

on images for textual content extraction. Because of this restriction, they cannot generate handwriting strokes for arbitrary text in a given calligraphic style unless provided with the text in the desired calligraphic style. In general, stroke prediction networks cannot generate text in unseen calligraphic styles because of their lack of textual conditioning and dependence on image features for both textual content and calligraphic style extraction. In our work, we present a handwriting stroke generation network that can generate strokes for an arbitrary text. The high-level workflow of the desired handwriting stroke prediction framework is shown in Fig. 1. The input is a calligraphic style in the form of an image and an arbitrary text, e.g., *"written by the same writer"*. The system outputs a new handwritten stroke sequence that imitates the given calligraphic style.

On the other hand, recent advancements in generative models for natural images [56, 64, 38] have facilitated handwriting image generation. The natural images are conditioned on text prompts, whereas, the handwriting images are conditioned on calligraphic style. However, the previous handwriting image generation methods use weak style learning networks to facilitate the readability of text [25, 26, 57]. These networks are based on GAN architecture and give less emphasis on calligraphic style features as compared to stronger textual features to increase the readability In contrast, our work proposes to use multi-scale attention-based features for a handwritten image. We have noticed that by using strong multi-scale calligraphic style features, our method performs well on unseen style images (see Section 6.2.3) while keeping the readability intact. Additionally, the image generation methods [25, 26, 57] learn the background texture. However, for handwriting image generation the handwriting style is more important than the background texture. One of the challenges handwriting images offer is the variation of calligraphic styles. Calligraphic style may include characters' shape and connectivity, font size, and writing tilt.

The current methods [25, 26, 57] do not consider character pair style (character connectivity), resulting in incomplete feature representation for handwriting style. Our proposed multi-scale style feature extraction is specifically designed to provide distinctive calligraphic features for stroke generation for arbitrary textual content for character shape and character pair style. The existing approaches [53, 45] for handwriting stroke generation often are unable to provide distinctive features for different styles, which in turn reduces the generation model's capability to mimic calligraphic style. In Table 1, we highlight the capabilities of the previous methods to mimic handwriting style. The handwriting stroke generation system is also preferred to work independently of the length of textual content. For instance, it is expected to generate readable strokes in a calligraphic style for short as well as longer text. As shown in Table 1 our method can effectively generate strokes in any desired style for short and long sentences.

Another issue that we address is the fact that the current image evaluation metrics are applied for the quantitative analysis of handwriting image generation. These metrics capture object diversity in generated natural images, but they are not designed to capture the calligraphic style and character shapes in handwriting. To evaluate the style similarity and character shape matching between generated and real handwriting images, we propose a style distance and projected character shape matching. To the best of our knowledge, ours is the first research to introduce evaluation metrics specifically capturing handwriting style and character shapes.

Chapter 1

Literature Review

1.1 Word detection

In the previous work, the words are segmented in the top-down approach where lines are segmented, then in the lines, the words and characters are segmented [47]. This approach uses a variable-sized window which is not robust to variation of word size in camera-captured images. [41] used confidence scores of word hypotheses from word recognition and lexical modeling to improve word detection. But if the lexicon method perform poorly then the word detection is also effected.

1.1.1 Scene text detection

Character detection shows promising results for natural image scene text detection. Nevertheless, the scene text is very different from the handwritten text. Words in natural image scene text are separated from one another in an arbitrary shape and mainly in a typed format. However, words in handwritten text often overlap with one another in adjacent lines. Furthermore, the spacing between characters in words may vary depending on the individual handwriting style. [69, 78] parameterized the word detection with the character gap, but these approaches are not effective against the overlapping words in handwritten text.

Moreover, scene text detection aims to localize the words in natural scenes from varying perspectives. However, in handwriting, the objective of word detection is to cope with different handwriting styles. Specifically, varying character/word spacing in handwritten text with uncontrolled camera conditions makes word detection challenging.

In the scene text detection domain, character detection is used to localize word instances. [40] construct word detector based on characters. The character region score detector is trained on word bounding boxes. Similarly, [7] localized the individual character and linked them to a text instance. The character region scores are trained in a weakly supervised manner with synthetic [29] and real dataset. Despite the success of character region scores in scene text detection, it cannot be directly applied to handwritten text since bounding box construction from character region scores cannot handle overlapping text frequently seen in handwriting.

Previous research explore various object detection frameworks for word detection in handwriting images.

1.1.2 Object detection

Another work [85] proposed to use a Cascade R-CNN [14] for handwriting detect. The invoice datsests is used in [85] with printed and handwritten parts. It detects the words from both handwritten and printed text. A two-stage framework is build in [83] where the first stage generates a region proposal for words and the second stage classifies the bounding box centered on a word. Also, [84] searched the word in historical handwritten documents by initializing the search using region proposals and embedding the proposals into word embedding space. These methods rely on a two-stage framework and proposal generation network. However, the presence of region pooling for region proposals in a two-stage network gives unsatisfactory results with handwriting images.

In recent years, the single-stage object detection algorithm has improved object detection accuracy and speed. In [52] a single-stage word detector [66] detect words and grade the examination automatically. In [70], [66] detect and recognize Kawi characters on copper inscriptions. These works are evaluated either on scanned images or printed text. However, the proposed framework for word detection is evaluated on more realistic low-contrast camera captures images.

The proposed word detector has the following contributions: 1) We explore the character region score for word detection in handwriting images. 2) Fuse the character region, affinity score, and input image for multi-channel word detection. 3) Our work is designed for low-contrast camera-captured handwriting images. 4) Our proposed character region score and input fusion outperform the state-of-the-art object detector for word detection in handwritten text. For a long time, handwriting analysis, such as handwriting recognition [21] and signature verification [20], has been an active research area.

1.2 Stroke trajectory recovery

There are two categories of handwriting, online and offline. Online handwriting is captured in real-time on a digital device such as a tablet screen with a stylus pen. In contrast, the handwritten text scanned or captured by a camera from a physical medium such as paper is referred to as offline handwriting [51, 60]. The handwriting inscribed on a digital device or captured from a physical medium is often unconstrained with varying orientations.

The availability of temporal movements of a stylus pen for online handwriting makes the handwriting analysis task easier. However, for offline handwriting, the input is limited to handwritten images, making handwriting analysis much more difficult.

The current STR architectures for English handwriting use lines of text [5, 11] or characters of alphabets [62, 63] as input. One of the recent datasets for STR, IAM-online [54], includes only line-level annotations. For English handwritten documents, line detection is a prominent topic for processing historical document images [13, 3]. A learning-free mechanism for detecting lines in a historical handwritten document is presented in [48]. Another algorithm for line segmentation in handwritten documents is proposed in [76]. This work is limited to separating the overlapping words only in horizontal lines [75]. For more complex Arabic handwriting, [24, 23] segment the curved text lines with overlapping words. A generative model to segment lines for Arabic handwriting is proposed in [18]. This method can segment slanting and curved lines but is not accurately enclosing a complete word in lines because they consider a binary mask to train the generative model. We have tried a few line segmentation [76, 3] methods. They are either proposed for non-English text or do not work well for unconstrained non-horizontal text in handwritten document images.

Non-English handwriting datasets [8, 88] emphasize word-level annotations and provide word, line, and page-level annotations for historical handwritten documents. The Chinese, Japanese, Arabic, and Tamil [11, 55, 79, 61] datasets for stroke trajectory also provide word-level annotations. Older STR datasets such as IRONOFF [80] consists of words/characters/digits of English handwriting. Unipen dataset [30] is available with character-level annotation, whereas IRONOFF consists of words. But we were unable to obtain any copies of IRONOFF datasets. In contrast, the publically available English handwriting dataset IAM-online [54] for STR includes line-level annotations with missing word-level annotations. Therefore, we propose constructing a large-scale word-level annotation for the IAM-online dataset in our work. In recent years conventional methods [20, 71] devised rule-based algorithms for signature/word trajectory recovery. Moreover, stroke trajectory recovery has progressed through deep neural networks. For stroke trajectory recovery, [11] introduces a first trainable convolutional network. This LSTM architecture learns strokes from Tamil scripts with Euclidean distance loss, making it hard to apply to long words with multiple strokes, such as English handwriting. Recently, [62, 63] introduced an attention mechanism to train the writing order recovery for characters. These attention networks are trained on characters with L1-loss, which is, again difficult to train for words. Similarly, [55] employs an LSTM architecture with an attention layer and Gaussian mixture model trained with cross-entropy loss. However, it is limited to encoding only a single Japanese character. Recently, [5] presents the stroke trajectory recovery, where LSTM is trained with a Dynamic Time Waring (DTW) loss function. [5] has a disadvantage that it can work only for a line of text. Apart from the restriction of the input to the lines of text, DTW loss has a drawback for long sequence matching: it sums the loss function for all the points when finding the best alignment between two sequences. Hence order preserving stray points, i.e., predicted stroke points far apart from their matching originals, have a minor influence on the DTW loss. However, they result in noticeable artifacts in the predicted strokes. To circumvent this issue, we propose to add the Chamfer distance [1] between predicted and ground truth point sets to the loss function. In order to prevent penalizing stroke points with small deviations, we augment the Chamfer distance to a marginal Chamfer distance. Applying the marginal Chamfer distance yields a more significant penalty for stray points/artifacts.

Another challenge a stroke trajectory recovery system faces is the availability of ground truth strokes. Annotating ground-truth strokes for the STR system is a laborious process that demands time and resource allocation. Therefore, our work proposes an evaluation algorithm that does not require ground truth stroke points to evaluate the STR system. To the best of our knowledge, the evaluation of the STR system without ground truth stroke points has not been proposed before.

1.3 Handwriting image generation from style image

The first few approaches for handwriting image generation conditioned on calligraphic style are based on GAN architecture. [4] proposed the first GAN-based architecture for synthesizing handwritten text images focusing on seen words only. Similarly, [43] proposed a few-shot style-conditioned handwritten word generation. However, it is limited to synthesizing short words rather than long texts. ScrabbleGAN [22] also synthesizes handwritten texts by concatenating all the letter tokens. It can be applied to any length of text but it does not generalize the calligraphic styles well and exhibits a lack of imitation ability. The above-mentioned GAN architectures are trained with multiple samples of images such as 15 samples with the same calligraphic style. Recent architectures utilize transformer encoder-decoder networks for handwriting image generation. The transformer architecture [10, 59] shows an advantage over LSTM models [5, 34] since they are not restricted to only predicting in forward direction.

In general, transformer-based handwriting imitation networks take a long time to train and are difficult to converge.

Recently, [25] has been proposed to utilize a GAN-based framework for handwriting image generation. It offers an advantage over the previous methods in that it can be trained using a single-word image as a sample. However, the generated images are slightly blurred and do not follow a unique character shape. To offer the image quality improvement to [25], [26] proposed to include a patch discriminator to the GAN architecture in [25]. The local patches of the image are fed into a patch discriminator and trained to improve the blur in the generated images. [26] can produce realistic and readable output by taking advantage of patch discriminator, text recognition, and writer identification modules. However, it overfits to text background. It seems to generate images with output even when there is no background texture in the style image (see Fig. 6.6). It is also not able to generalize to unseen styles and requires a large memory to train because of several auxiliary networks. [57] presents the latest diffusion model-based image generation network without auxiliary networks. The iterative learning of the diffusion model also requires a long time to train and often does not generate readable output. This network is trained with writer class information hence it cannot be applied to unseen styles outside the dataset. Additionally, these methods [25, 26, 57, 10, 59] only generate words and are not able to process long sentences and exhibit limited applicability for unseen styles.

Furthermore, for handwriting image generation, the learning network aims to generate images directly from calligraphic style features [16, 17, 42]. However, the image generation method requires a long time to train [57]. On the other hand, handwriting stroke generation networks constitute fewer parameters to train and, therefore, can be computationally much faster to train [53, 5, 34].

1.4 Handwriting strokes generation from style image

The proposed approach belongs to handwriting stroke generation methods. In contrast to handwriting stroke prediction and image generation, these methods can generate strokes for an arbitrary textual string in any arbitrary calligraphic style. The handwriting stroke generation has an advantage over the image generation. A stroke generation network deals with fewer parameters as compared to an image generation network and requires less time to train the network.

[45] decouples the textual and style features from handwriting images. It proposes an implicit learning of textual and style features simultaneously. However, it does not generate readable text because of the imperfect separation of textual and calligraphic features. Most recently, [53] proposed a method to predict strokes from handwriting images. It employs a diffusion model conditioned on text and style features to generate the stroke sequence for any arbitrary text in a calligraphic style. It consists of a diffusion model without auxiliary networks, which can be trained in a reasonable time. However, it fails to generate strokes in the same calligraphic style and lacks diversity in the generated stroke styles. The drawback of this model is that they used mobile net [39] trained on natural images to extract the features from handwriting images. It results in irrelevant features for handwriting and does not represent the diversity in handwriting calligraphic styles since the generic feature extraction

network does not represent handwriting features. Thus, it cannot mimic the calligraphic style of the image. In the next section, we present our proposed diffusion model trained with strong style features and Dynamic Time Warping loss.

1.5 Contribution

Overall, handwriting stroke prediction and generation for arbitrary textual content and calligraphic styles is a challenging problem. Our work makes the following main contributions:

- We introduce large-scale word-level annotations for the English handwriting STR dataset sampled from the IAM-online dataset. Our version of the IAM-online dataset contains 62,000 words.
- A word-level STR method estimates loss for each word rather than averaging DTW loss over the entire line of text. To avoid the stray points/artifacts in predicted stroke points, we employ a marginal Chamfer distance that penalizes large, easily noticeable deviations and artifacts.
- We also introduce an algorithm for evaluating the STR system on images without ground truth stroke points.
- Since our method works with words, we demonstrate that our method is scalable to unconstrained handwritten documents, i.e., full-page text.
- We propose a simple probabilistic diffusion model for handwriting stroke generation. As opposed to handwriting generation as images, our model is small and efficient to train.
- We propose multi-scale attention features to represent the character embedding.
- We also propose a character pair embedding to represent the connectivity between characters based on multi-scale features.
- We trained our diffusion model with Dynamic Time Warping (DTW) and diffusion loss to improve the stroke prediction.
- We introduce new metrics to evaluate calligraphic style similarity that utilize style features and character shapes.
- Our system can compete with image generation in terms of calligraphy style imitation via plotting strokes as images. Our quantitative and qualitative results show our proposed method's effectiveness and generalization ability.

Chapter 2

Word Detection in Camera-captured Handwriting Documents

2.1 Method

In our work, we propose to fuse character region and affinity score with the input image to determine the word localization for camera-captured handwritten text. The character region and affinity scores give information about the character's existence and probability of character belonging to the same word, respectively. Characters are fundamental building blocks for camera-capturing handwritten images. However, the handwritten text lacks character annotation for words. So, we adapt the character region score from [7]. In [7], an encoder-decoder pair is trained for character region score with 80k synthetic images having character bounding box annotations. Encoder in character region score network consist of VGG-16 [73] backbone and decoder with skip connections similar to U-Net [68]. The encoder-decoder pair learns character region and affinity scores for handwritten text. The labels for characters in the handwritten text are generated in a weakly supervised manner. The predicted character bounding boxes from handwritten text and ground-truth character bounding boxes from synthetic datasets [29] are used to train the encoder-decoder pair, which predicts character region scores for handwriting images. The loss function for character region score and affinity map is given in eq. 4.1.

$$L = \sum_{p} S_c(p) \cdot \left(\|S_r(p) - S_r^*(p)\|_2^2 + \|S_a(p) - S_a^*(p)\|_2^2 \right) \tag{2.1}$$

where $S_r^*(p)$ and $S_a^*(p)$ denote the pseudo-ground truth region score and affinity map, respectively, and $S_r(p)$ and $S_a(p)$ denote the predicted region score and affinity score, respectively.

The two fundamental challenges in designing a word detection algorithm for practical application are the diversity in handwriting styles and the low-contrast of camera-captured

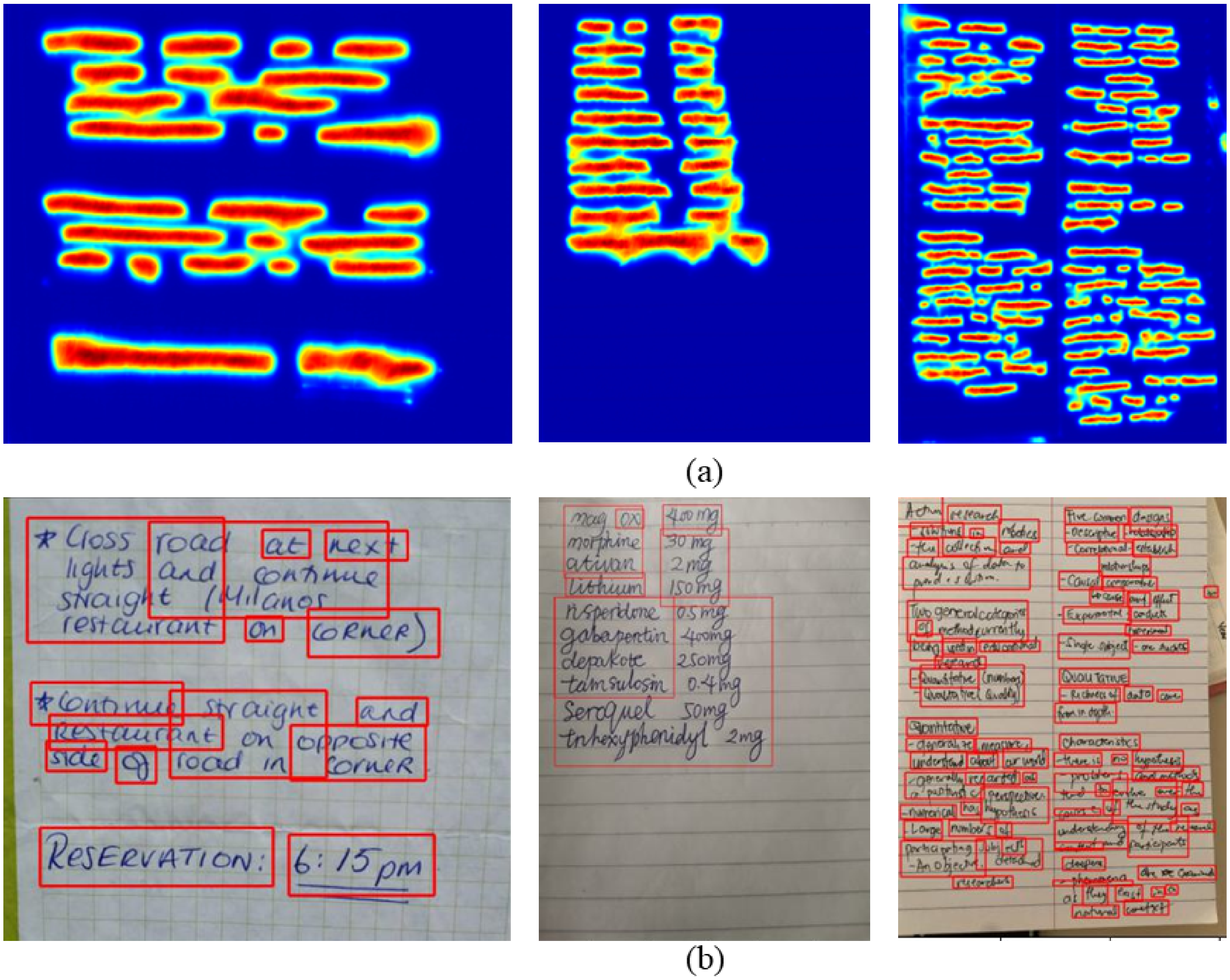


Figure 2.1: (a) character region + affinity score (b)word detection based on (a)

handwriting images. Fig. 2.1(a) shows the normalized sum of character region scores and affinity scores for GNHK handwriting datasets [49]. The character map predicted by weakly supervised learning is a good indicator of the character's presence. However, Fig. 2.1(b) shows that the deterministic method to construct bounding boxes on these character region scores [7] is not able to detect the word for handwritten text. Therefore, in our work, we propose to train a multi-channel object detector with these character region scores described in the Sec 2.1.1.

2.1.1 Multi-channel word detector

Though character region scores are a good indicator for words, as shown in 2.2(a), the bounding box estimation on character region scores cannot handle a handwriting style. Fig. 2.2(b) highlighted that character region scores alone are insufficient to perform well for overlapping words in adjacent lines.

To keep the advantages offered by the character region score and to prevent the problem shown in 2.2(b), we propose to fuse the character region and affinity scores with the input image to learn word detection from handwritten text. Word detection from handwritten text is a single-class detection problem with multiple steams of input information. The detection network consists of convolutional layers with dense connections and pyramid pooling. It is a multi-channel word detection framework for handwritten text. In previous research, multi-channel input is utilized for object detection in satellite images [77] and outdoor scenes [81]. However, these researches used additional information bands such as infrared or depth maps, which are readily available with datasets. However, we propose learning the character region score in weakly supervised manner without any character level annotations. We fuse the character region score and affinity scores for word detection into an object detection framework [67, 12, 35]. Single-stage objects detector [67] performs better on word detection than a two-stage object detectors [12]. The work in [6] used region score to detect word. It estimates the heat map of words and generates the region proposal on the estimated heat map. The heat map and regional proposals are fed into the filter network to learn if the region proposal envelops a word. This work is very different from our proposed approach. First, it does not learn any information about the character regions and is limited to estimating word region scores. Secondly, their region proposal generation is also limited to a heat map of words. However, in our work, we propose combining the cons of both handwriting image and its character region scores. The character region scores break the word entity into basic building blocks (character), and the affinity map provides the probability of them belonging to the same word. Our proposed method is independent of vocabulary and considers words' character region and affinity scores in handwritten text. Therefore, it can be easily scalable

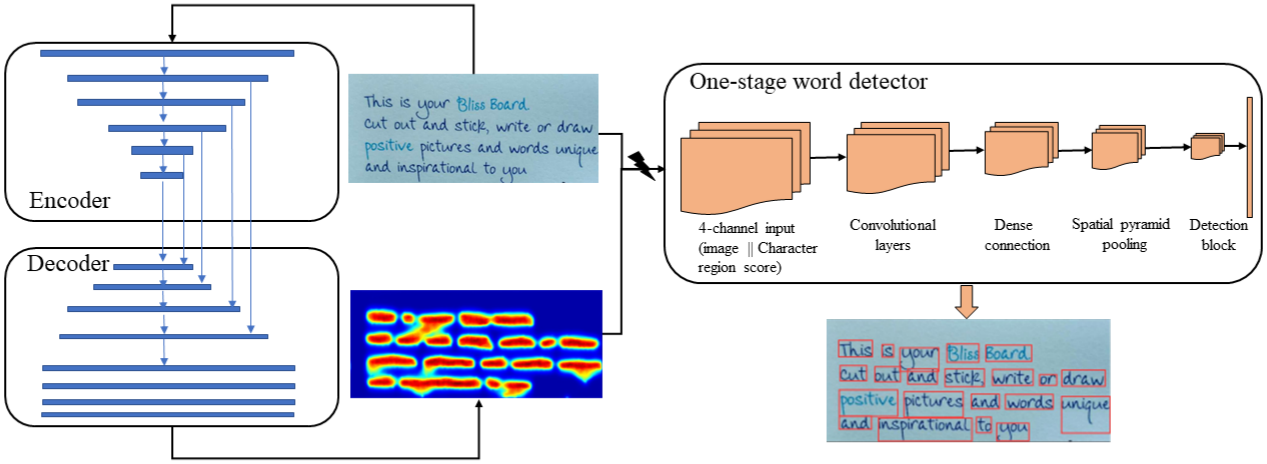


Figure 2.2: The block diagram of the convolution network comprising encoder, decoder and detection branches. Encoder-decoder pair is used to learn character region scores and detection branch learn the multi-channel word detection for fused character region scores and handwriting image. The break symbol between encoder-decoder branch and detection branch shows the autonomy of both branches.

Method	mAP@0.5	mAP@0.5:0.95
Two-stage object detector [67]	78.0	56.5
Character region score [7]	60.3	56.5

Table 2.1: The detection accuracy for two-stage object detector [67] and Character region score [7]

to any document type and vocabulary.

2.2 Experiments

In the next section, we briefly describe the low-contrast camera-captured GNHK dataset [49] and evaluate the performance of word detection on it.

2.2.1 Datasets

The images in GNHK datasets are sourced from Europe, North America, Africa, and Asia. It is a diverse dataset as penmanship varies in different parts of the world. The dataset consists of 687 images containing different types of handwritten text, such as shopping lists, sticky, and diaries notes. Mobile phone cameras captured images under unconstrained settings. Therefore, it may contain shadows from mobile devices, and handwriting has very low contrast with the background, as visible in Fig. 2.3. There is a corresponding JSON file for each handwritten-text image containing the annotations of words in the images. Fig. 2.3(a) shows the image of handwritten text, and Fig. 2.3(b) shows the ground truth bounding boxes of words for each word in handwritten text.

2.2.2 Results and discussion

In our work, a multi-channel object detector is proposed for word detection for GNHK datasets [49]. In [49] the baseline is established with two-stage word detector [67, 82]. Table. 2.1 shows the performance of the two-stage object detector and character region score. It can be seen in Table. 2.1 that the object detector and character region score have the same mAP@0.5:0.95 accuracy (0.565), however for mAP@0.5 two-stage object detector (0.780) have higher accuracy than character region score (0.603). The low accuracy of the character region score is because of a deterministic bounding box estimation for words [7]. For overlapping text in handwriting, the deterministic estimation does not work for handwriting text, as shown in Fig. 2.2.

In our work, we propose to perform multi-channel word detection leveraging character region and affinity score along with handwriting text image. The single-stage object detector outperforms the two-stage detector by a large margin. Table. 2.2 shows the quantitative

Input	Prec	Recall	mAP@0.5	mAP@0.5:0.95
Word detector [12]				
image	90.1	87.2	91.8	63.6
Multi-channel word detector with character region scores [12, 7]				
image —— RS	**90.3 (+0.2)**	87.7	91.8	64.0
image —— (RS + AS)	89.8	**88.4 (+1.2)**	**92.2 (+0.4)**	**64.0 (+0.4)**

Table 2.2: The detection statistics for image, character region and affinity score on multi-channel object detector network [12]. In the table, image stands for 3-channel handwriting image, *RS* stands for character region score and *AS* stands for character affinity scores.

Handwriting size	Prec	Recall	mAP@0.5	mAP@0.5:0.95
image				
Large	82.0	89.6	89.5	64.8
Med	91.5	**92.2**	94.1	**68.0**
Small	**89.5**	86.1	91.1	**62.2**
image —— (RS + AS)				
Large	**84.9**	**90.7**	**91.1**	**65.7**
Med	**92.0**	91.7	**94.4**	67.1
Small	89.2	**86.9**	**91.3**	62.1

Table 2.3: The detection statistics for image and character region scores as input for the object detector network [12] for large, medium, and small handwriting sizes

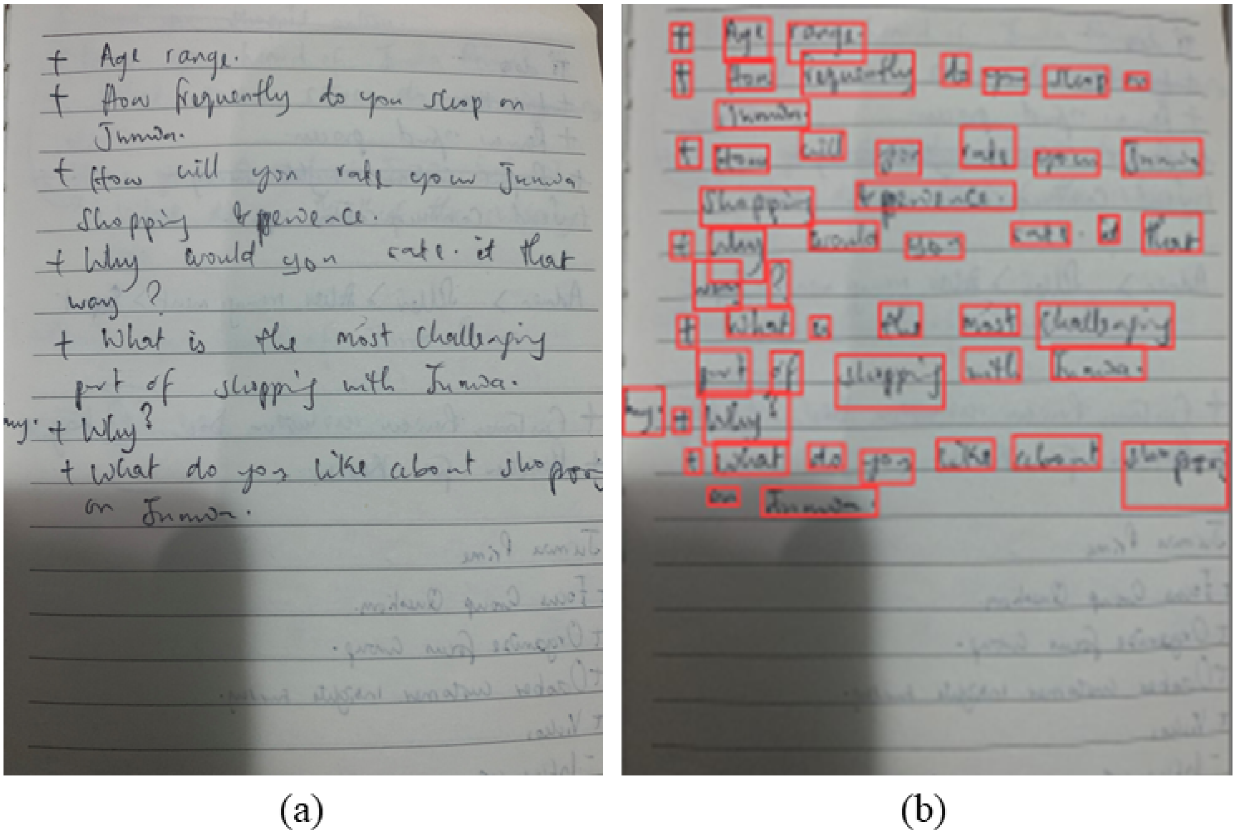


Figure 2.3: Sample images from GNHK datasets [49] with bounding boxes

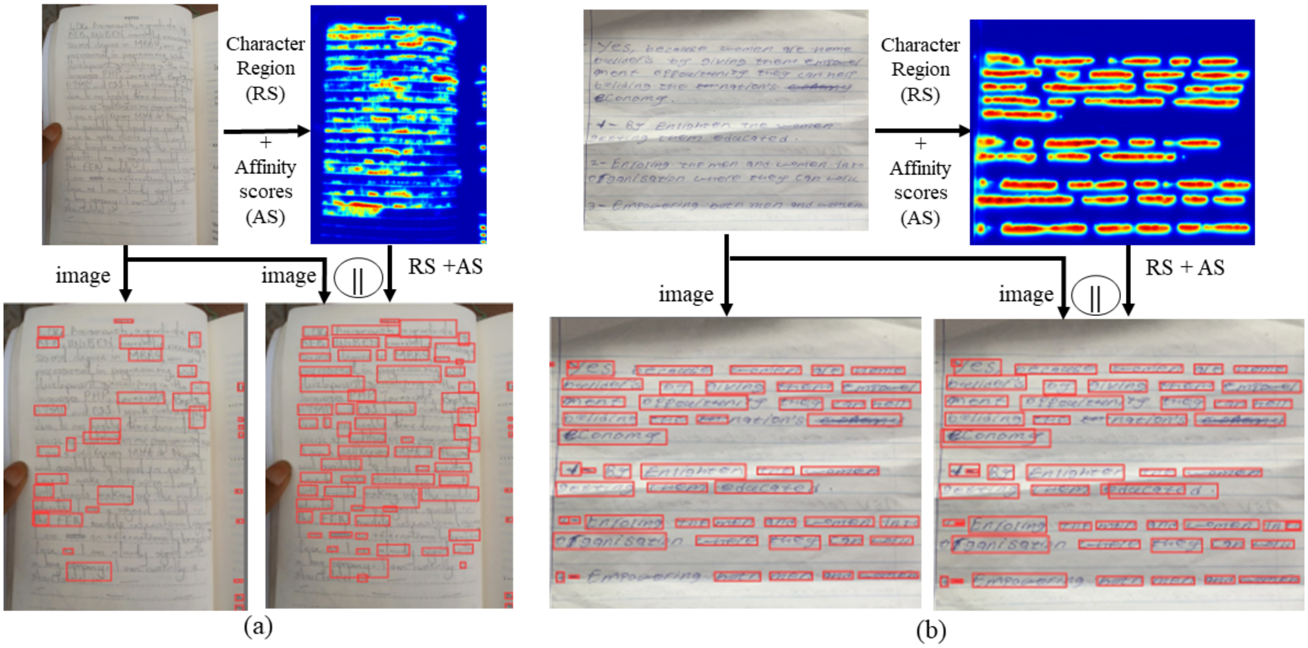


Figure 2.4: The qualitative results for word detection for low-contrast camera -captured handwritten text

results for image, character region (RS), and affinity scores (AS) for the word detector network [12]. In Table. 2.2, we can see that the multi-channel information consisting of handwriting image, character region scores, and affinity map outperforms the word detector without character region and affinity scores [12]. The multi-channel word detector increases the recall by 1.2%, mAP@0.5, and mAP@0.5:0.95 by 0.4%. Therefore, additional information from weakly supervised character region scores beats the state-of-the-art word detector.

Fig. 2.4 show the qualitative results for challenging examples with low-contrast and overlapping words. Nevertheless, our method still produces a reasonable detection compared to state-of-the-art word detector [12]. In that case, the character region and affinity score provide the word clues shown in character scores map in Fig. 2.4.

We also validated in Table. 2.3 that character region and affinity scores gives better performance for large and medium-size words. It gives approximately 1% improvement in mAP@0.5 with the single-stage word detector. On the other hand mAP@0.5:0.95 declines for small handwriting text as the quality of character region scores declines for very small word sizes. In Table 2.3, we illustrate the results for large, medium, and small handwritten text. Character region and affinity scores for large and medium handwriting outperform input images for word detection. Character region and affinity scores as word clues with RGB images of handwritten text improves the word detection accuracy on camera-captured handwriting images.

2.3 Conclusion

In our work, we propose the multi-channel word detection that leverage the character region scores trained in a weakly supervised manner for handwritten text. The character region and affinity scores improve the qualitative and quantitative results. The state-of-the-art word detector struggles to detect words in low-contrast camera-captured handwriting images. However, the proposed multi-channel word detector performs well also on challenging examples.

Chapter 3

Strokes Trajectory Recovery for Unconstrained Handwritten Documents

3.1 Method

3.1.1 Network architecture

We deploy a CNN with bidirectional LSTM for stroke trajectory learning. The input to the CNN is a word image resized to a fixed height with variable width to keep the same aspect ratio. A CNN branch consists of seven convolutional blocks with ReLU activation. The convolutional filters have a 3x3 kernel size with 2x2 and 2x1 max pooling in each layer. The output of the CNN branch, a Wx1024 dimensional feature vector, where W is the width of the image, is fed into eight bidirectional LSTM blocks. Each bidirectional LSTM block consists of 128 hidden units, so each LSTM block's input is Wx128. Lastly, a bidirectional LSTM is followed by a 1-D convolutional block that predicts a Wx4-dimensional output. The number of output points is proportional to the width of the input image. The first two dimensions of the output indicate the relative coordinates (x,y) with respect to the previous location. The last two dimensions indicate start-of-stroke (*sos*) and end-of-stroke (*eos*) tokens, respectively. Cross-entropy loss is employed to learn the start-of-stroke and end-of-stroke tokens. The overall architecture is shown in Fig. 6.1.

3.1.2 Loss function

The ground truth in IAM-online dataset [54] is a sequence of points defined as (x, y) coordinates with a time stamp. Since the number of predicted coordinates is propositional to the width of the input image, we re-sample the equidistant ground-truth coordinates such that the number of points is proportional to the image's width.

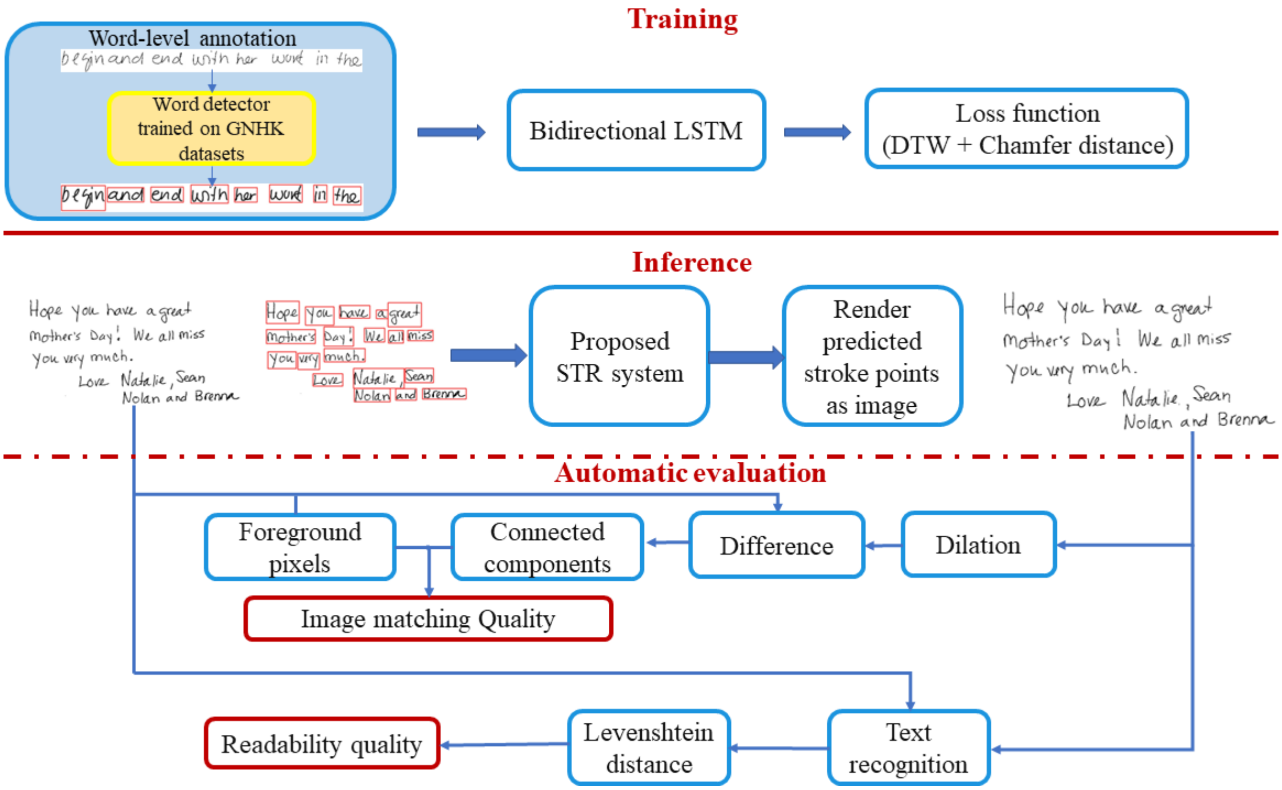

Training
Word-level annotation
begin and end with her work in the
Word detector trained on GNHK datasets
begin and end with her work in the
Bidirectional LSTM
Loss function (DTW + Chamfer distance)
Inference
Hope you have a great Mother's Day! We all miss you very much. Love Natalie, Sean Nolan and Brenna
Proposed STR system
Render predicted stroke points as image
Automatic evaluation
Foreground pixels
Connected components
Difference
Dilation
Image matching Quality
Readability quality
Levenshtein distance
Text recognition

Figure 3.1: The block diagram of the overall system for training, inference, and automatic evaluation.

Dynamic Time Warping (DTW) loss

In general, DTW [9] computes the optimal match between ground-truth (GT) ($T = (t_1, t_2, t_3, ...t_m)$) and predicted sequences ($P = (p_1, p_2, p_3, ...p_n)$) of different lengths by finding the warping path between two sequences. In DTW loss, the cost matrix is calculated as:

$$cost(i,j) = ||p_i - t_j||^2 \tag{3.1}$$

The accumulative cost matrix (A) is given as,

$$\begin{aligned} A(i,j) = cost(i,j) + min[A(i-1,j), A(i-1,j-1), \quad (3.2)\\ A(i,j-1)] \quad (3.3) \end{aligned}$$

for $1 \leq i \leq n$ and $1 \leq j \leq m$.

Given matrix A, DTW computes the optimal warping path from $A(n,m)$ to $A(1,1)$ as the alignment of points in P to points in T is expressed as index mapping $\alpha : \{1, \ldots, n\} \rightarrow \{1, \ldots, m\}$, where α is an onto function.

$$\mathcal{L}_{DTW}(P,T) = \sum_{i=1}^{n} ||p_i - t_{\alpha(i)}||. \tag{3.4}$$

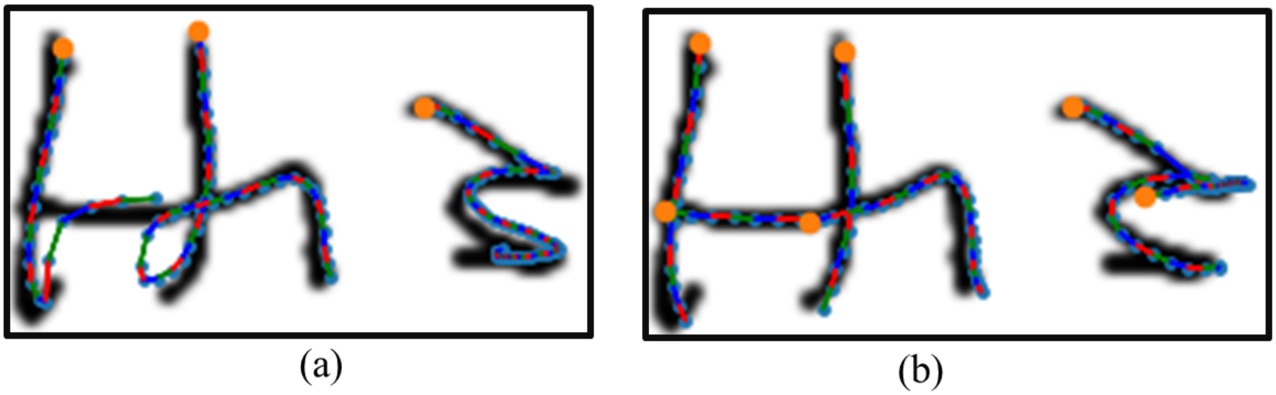


Figure 3.2: The impact of marginal Chamfer distance loss on the stroke trajectory recovery. (a) DTW loss only. (b) DTW + Chamfer distance loss.

Chamfer distance loss

The DTW gives promising results for training the stroke trajectory recovery systems [5]. It helps to match the point trajectory for ground truth and predicted strokes. However, the DTW loss function does not impose a sufficiently large penalty for predicted points following a similar trajectory as the ground truth points but far off compared to the ground truth at the pixel level. Fig. 3.2(a) shows that the predicted and ground truth points are far off, but the DTW loss is small since the predicted strokes follow the same trajectory as a ground truth stroke. Therefore, in this work, we propose to add a marginal Chamfer distance. Its effect is illustrated in Fig. 3.2(b). The proposed marginal Chamfer distance between the predicted and ground truth point sets is given by

$$d_{CD}(P,T) = \sum_{p \in P} \max(\min_{g \in T} \|p - g\|_2^2) - c^2, 0) + \tag{3.5}$$

$$\sum_{g \in T} \max(\min_{p \in P} \|g - p\|_2^2) - c^2, 0), \tag{3.6}$$

where P and T represent the point sets for predicted and ground truth strokes. The Chamfer distance is calculated on the pixel level. In the experimental section, we will discuss the setting of parameter c. The intuition behind the marginal Chamfer distance for STR is to consider the distance only if the predicted stroke point is at least a unit pixel apart from its nearest ground-truth stroke point. It leads to quantitative improvements in loss calculations as shown in Table 5.1. The proposed loss is simply a sum $\mathcal{L}_{DTW}(P,T) + d_{CD}(P,T)$

3.2 Experiments

We first present the construction of word-level annotations for IAM-online dataset (Sec. 3.2.1) and introduce the greeting-card handwritten messages (GHM) dataset (3.2.1). We

Method	mAP@0.5	mAP@0.5:0.95
Scene text detector [7]	0.603	0.565
Two-stage word detector [67]	0.780	0.565
Single-stage word detector [12]	**0.913**	**0.619**

Table 3.1: The accuracy (mAP) of word detection with state-of-the-art scene text detector [7] and single and two-stage object detectors [67, 12].

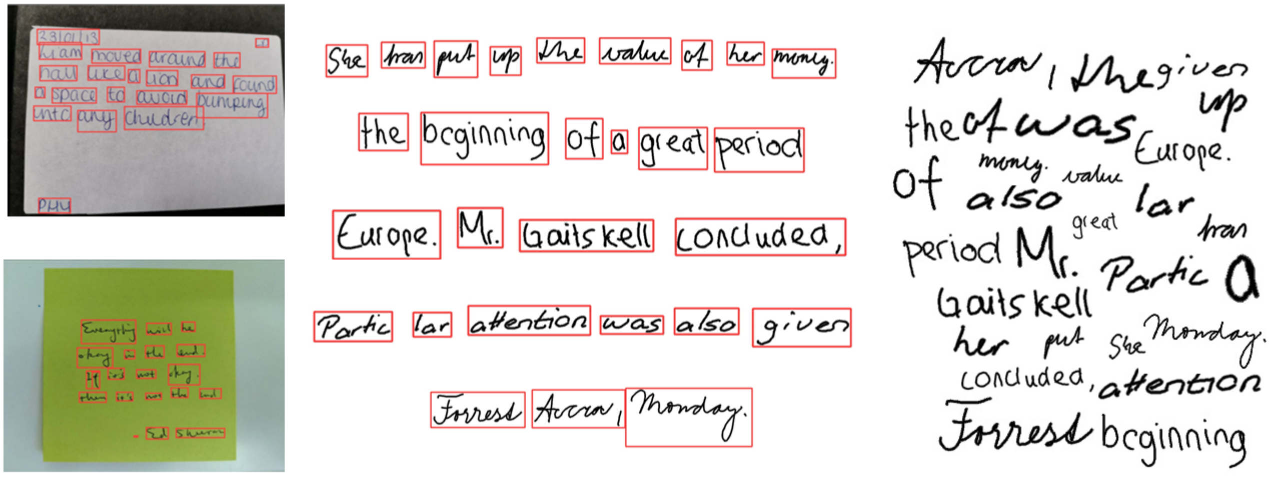
discuss the evaluation metric and results of the proposed method on these datasets (Secs. 3.2.2 and 6.2.3). Finally, we present the application of our proposed method for the GHM dataset (Sec. 3.3.2).

3.2.1 Word-level annotation

In the IAM-online dataset [54], the stylus pen movement on an electronic device's screen provides the coordinates of the ground-truth point for stroke trajectory recovery. However, the IAM-online dataset only provides the stroke's ground truth for the line-level text. Therefore, we use word detection to construct stroke annotations for words. Word detection generally requires word bounding boxes to train the detection network, but the IAM-online dataset does not include word bounding boxes. So, we propose to train the word detection on GNHK [49] dataset and then applied to the IAM-online dataset. The images in the GNHK dataset are sourced from different regions of Europe, North America, Africa, and Asia containing 39k+ words, sufficient to train a word detector with data augmentation [12]. Hence, it is a diverse dataset regarding writing style and image quality as the penmanship varies in different parts of the world, and camera quality varies for each captured image.

We explore three state-of-the-art detectors; one scene text detector [7] and two object detectors for localizing the words in unconstrained handwritten text. A scene text detection network identifies character regions in natural images and detects the words based on character regions and affinity scores between them [7]. However, the deterministic approach to constructing bounding boxes around character region scores is not well suited for handwriting word detection. Because in the handwritten text, adjacent lines may overlap, and characters may have high affinity scores in adjacent lines, which misleads the word detection results. This shortcoming of the scene text detector gives low detection accuracy for unconstrained handwritten text (as shown in Fig. 3.4).

On the other hand, the object detector attributes words as objects and is more efficient for detecting overlapping words in adjacent lines in a non-horizontal orientation. For our task, a single-stage detector YOLO [12] performs better than a two-stage detector Faster R-CNN [67, 36, 33]. Both are trained and evaluated on GNHK dataset [49]. In Table 3.1, we list the quantitative results for word detection on GNHK dataset. Due to its best performance, we select the single-stage YOLO word detector [12] and apply it to IAM-online dataset [54] for word detection.


She tran put up the value of her money.
the beginning of a great period
Europe. Mr. Gaitskell concluded,
Partic lar attention was also given
Forrest Accra, Monday.

(a) (b) (c)

Figure 3.3: (a) Word detection for the GNHK dataset [49], (b) word detection for IAM-online dataset [54], (c) Words from IAM-online dataset used to train our network.

In Fig. 3.3(a,b), we show the word detection visualization for sample images from GNHK dataset [49] and IAM-online dataset [54] respectively. The images in GNHK [49] vary in handwriting style, background, and camera conditions. Therefore, the word detection trained in GNHK dataset [49] has acceptable performance for the IAM-online dataset [54]. We used the word-level annotations for the IAM-online dataset [54] to finetune and evaluate our system. Word detection on the IAM-online dataset gives us 41,665 and 19,496 words for train and test sets, respectively. Fig. 3.3(b) shows the word detection from lines from IAM-online, and Fig. 3.3(c) shows the words used in our work to train the network.

Greeting card messages dataset

In our work, we propose a word-level stroke trajectory recovery.

To evaluate our system on unlabelled handwritten documents, we acquire approximately 2,000 greeting-cards handwritten messages (GHM) dataset from greeting cards company [72]. The GHM dataset shared[1]. Handwritten messages from the GHM dataset do not follow any fixed template because it consists of user-uploaded handwritten messages to greeting cards company [72]. The samples from the GHM dataset are shown in Fig. 3.4.

3.2.2 Evaluation metric

We used a distance-based evaluation metric to evaluate stroke trajectory recovery. The average distance of points in the ground-truth (T) stroke to its nearest predicted stroke (P)

[1] https://drive.google.com/file/d/1G-EZBfEhsHThR9dR1YtJdPE3Mg0ay0w-/view?usp=sharing

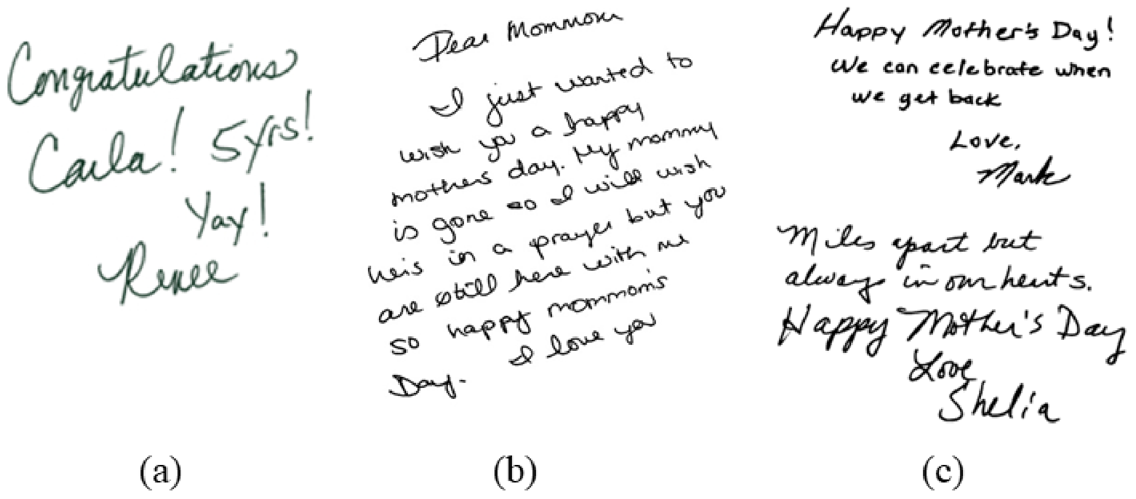


Figure 3.4: Sample documents form GHM dataset with unlabelled handwriting images.

Loss function	$dist_{t,p}$ **(mean)**	$dist_{t,p}$ **(std)**	$dist_{p,t}$ **(mean)**	$dist_{p,t}$ **(std)**	ϵ_{sos}
DTW_{line} [5]	0.01558	0.01402	0.02776	0.0353	0.1553
DTW_{word}	0.01653	0.014862	0.01287	0.00965	**0.1427**
DTW_{word} + Chamfer distance	**0.01492**	**0.01042**	**0.01257**	**0.00946**	0.1479

Table 3.2: The quantitative comparison for training on DTW loss for lines and words. The third row lists the quantitative results for training with combined DTW and Chamfer distance loss between predicted and ground-truth points

is denoted by $dist_{t,p}$. Similarly, the average distance of points in the predicted stroke (P) to its nearest ground-truth stroke (T) is denoted by $dist_{p,t}$. The metric $dist_{t,p}$ signifies that every ground-truth stroke point is close to the predicted point and vice versa for $dist_{p,t}$. $dist_{t,p}$ and $dist_{p,t}$ are the same evaluation metrics as used in [5]. However, apart from the mean (*mean*) of the distances between predicted and ground truth stroke points, we also compute the standard deviation (*std*) of the metrics. We also compute the loss for predicting the start-of-stroke token ϵ_{sos}. ϵ_{sos} should have the lowest value if the start-of-stroke token is predicted correctly.

Loss function		$dist_{t,p}$ **(mean)**	$dist_{t,p}$ **(std)**	$dist_{p,t}$ **(mean)**	$dist_{p,t}$ **(std)**
DTW_{word} + Chamfer distance	c = 1	0.01492	0.01042	0.01257	0.00946
	c = 3	0.0177	0.0118	0.01859	0.0162

Table 3.3: The quantitative comparison of increasing the value of c from 1 to 3.

3.3 Results

DTW and Chamfer distance are complementary loss functions to train our system. DTW loss ensures that the predicted stroke sequences are similar to the ground truth stroke sequence. The Chamfer distance between the predicted and ground truth point set ensures that there are no spurious points/artifacts in predicted points; that is, no predicted points are far away from ground truth points.

3.3.1 IAM-online dataset

Table 5.1 presents a quantitative comparison, where bold numbers show the best results (lowest value). The first and second rows of Table 5.1 show the evaluation with line-level and word-level input for DTW loss, respectively. We observe that both $dist_{t,p}$ and $dist_{p,t}$ metrics are much lower for word-level than the line-level input. These results show that training the stroke trajectory recovery with a word-level dataset, as we proposed, improves the results. We also validated that Chamfer distance loss for predicted and ground truth point sets improves the quantitative results. Both mean and standard deviation of $dist_{p,t}$ and $dist_{t,p}$ decrease by adding Chamfer distance to the loss function. It means that the predicted strokes are better at imitating the ground-truth strokes. The lower values of both $dist_{t,p}$ and $dist_{p,t}$ illustrate that every ground-truth stroke has a close predicted stroke and vice versa. So, we do not get spurious predicted strokes and yet do not miss to follow the shape of the ground-truth strokes. We noticed that chamfer distance loss has minimal influence on start-of-stroke (ϵ_{sos}) as shown in Table 5.1. In another experiment, we try the higher values of c (Eq. 3.6) as shown in Table 3.3, but by increasing the value of c increases the *std* of $dist_{t,p}$ and $dist_{p,t}$. Therefore, we keep the value of c=1 in our work.

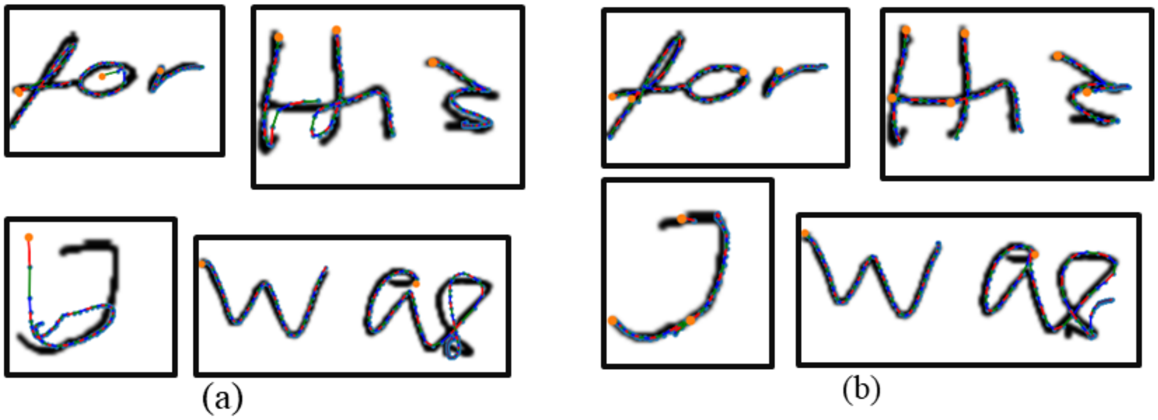


Figure 3.5: Samples of stroke trajectory recovery with (a) DTW loss and (b) DTW loss with Chamfer distance on the predicted and ground truth point sets. The recovered stroke trajectory is shown by red, blue, and green arrows, and the predicted start-of-stroke point is shown as an orange circle (best view in colored).

The visualization of the elimination of spurious predicted points/artifacts from the predicted stroke trajectory by adding Chamfer distance loss is illustrated in Fig. 3.5. We can see that extra stroke points cause more artifacts in (a) than in (b).

3.3.2 Greeting cards messages dataset

The previous methods on the IAM-online dataset work with line-level text for stroke trajectory recovery [5], which is not scaleable to unconstrained handwritten text images without line detection. This is one of the main reasons why we work with word-level text.

In one of the applications of the proposed work, we show the stroke trajectory recovery for greeting-card handwritten messages [72]. We applied the developed method trained for word-level annotation using DTW and Chamfer distance to the images containing greeting-card handwritten messages. Handwriting images from greeting card messages do not follow any fixed template because they consist of user-uploaded handwritten messages. Therefore, first, we detect the words in greeting card handwritten messages with word detector described in Sec. 3.2.1. Then we execute the trained STR model on each detected word. The recovery of stroke trajectory for greeting card handwritten messages is the word-level stroke trajectory recovery of each word.

We render the image from predicted stroke points as described in Fig . 3.6, where the proposed STR system predicts the stroke trajectory recovery for each word. Whereas the render image module converts stroke points into an image. Finally, we align the rendered image to the location of the detected word.

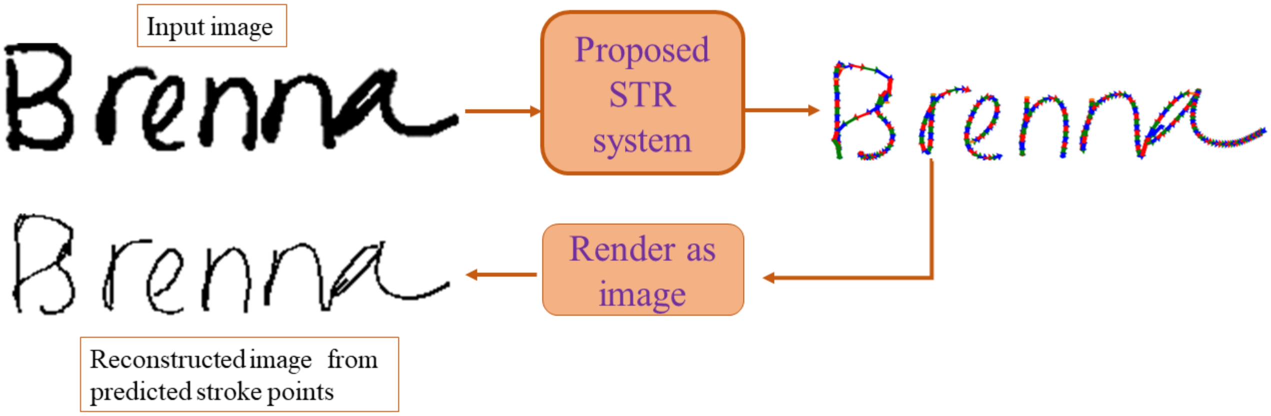
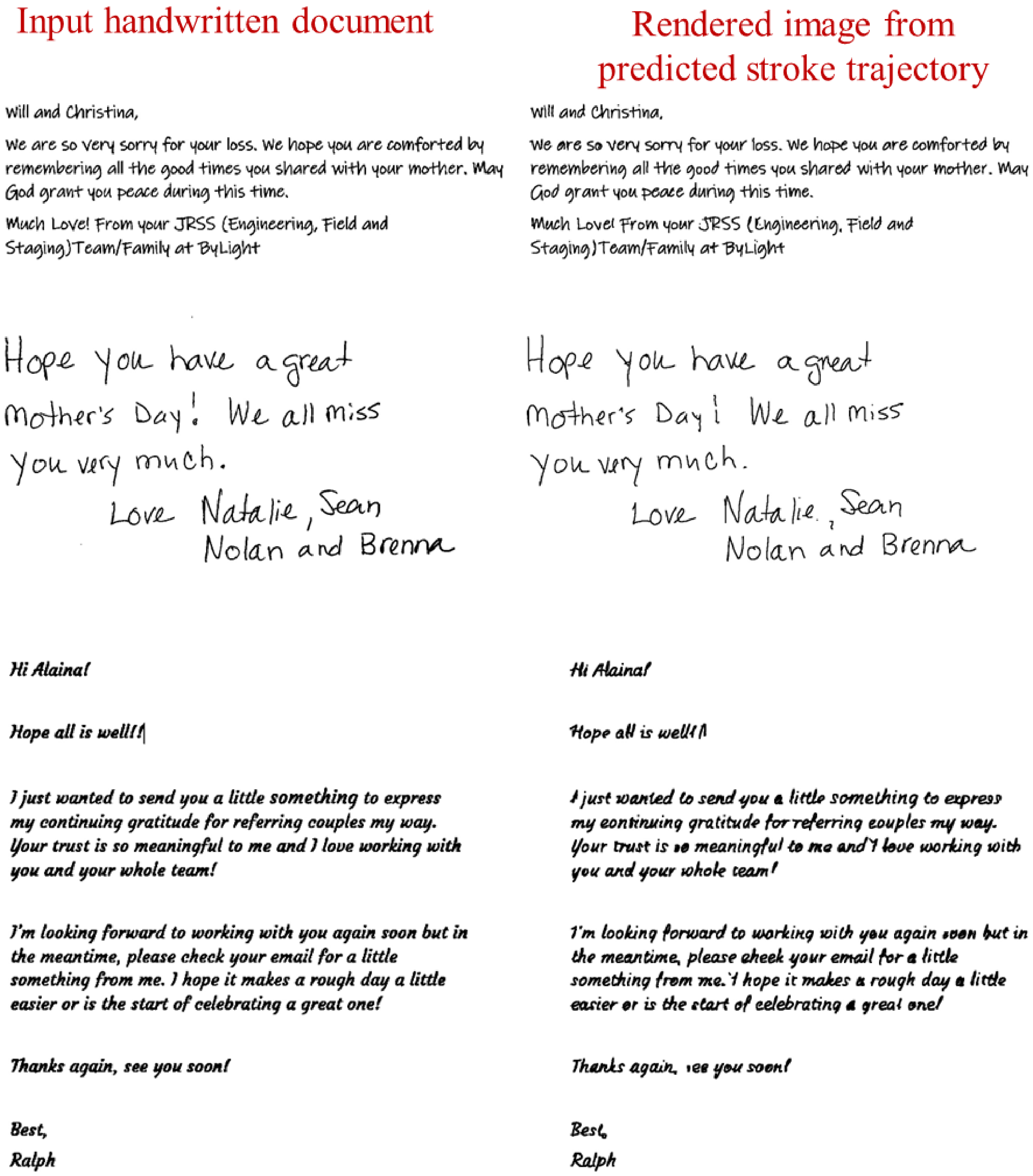


Figure 3.6: The mechanism of image rendering from predicted stroke points.

The visual results of the proposed word-level STR system on handwritten greeting card messages as shown in Fig. 3.7 are rendered word-wise by the image render module illustrated in Fig. 3.6. Our proposed STR system can easily be applied to text with different orientations, as shown in Fig. 3.7. The left-hand images are input handwritten messages, and the right-hand images are rendered from the recovered stroke trajectory for each word.


Input handwritten document
Rendered image from predicted stroke trajectory

Will and Christina,

We are so very sorry for your loss. We hope you are comforted by remembering all the good times you shared with your mother. May God grant you peace during this time.

Much Love! From your JRSS (Engineering, Field and Staging)Team/Family at ByLight

Will and Christina,

We are so very sorry for your loss. We hope you are comforted by remembering all the good times you shared with your mother. May God grant you peace during this time.

Much Love! From your JRSS (Engineering, Field and Staging)Team/Family at ByLight

Hope you have a great
mother's Day! We all miss
you very much.
Love Natalie, Sean
Nolan and Brenna

Hope you have a great
mother's Day! We all miss
you very much.
Love Natalie, Sean
Nolan and Brenna

Hi Alaina!

Hope all is well!!

I just wanted to send you a little something to express my continuing gratitude for referring couples my way. Your trust is so meaningful to me and I love working with you and your whole team!

I'm looking forward to working with you again soon but in the meantime, please check your email for a little something from me. I hope it makes a rough day a little easier or is the start of celebrating a great one!

Thanks again, see you soon!

Best,
Ralph

Hi Alaina!

Hope all is well!!

I just wanted to send you a little something to express my continuing gratitude for referring couples my way. Your trust is so meaningful to me and I love working with you and your whole team!

I'm looking forward to working with you again soon but in the meantime, please check your email for a little something from me. I hope it makes a rough day a little easier or is the start of celebrating a great one!

Thanks again, see you soon!

Best,
Ralph

Figure 3.7: The input image and the rendered image from predicted stroke trajectory recovery for GHM dataset.

3.4 Conclusions

In our proposed work, we trained a neural network with word-level annotations for the IAM-online dataset using DTW and Chamfer distance loss functions. We demonstrated that adding Chamfer distance loss to DTW is beneficial for removing artifacts and spurious stroke points for better stroke trajectory recovery. We also proposed automatic evaluation methods using image matching and readability consistency to evaluate the quality of stroke trajectory recovery for unlabeled datasets. Finally, we demonstrate the ability of our proposed work to work in unconstrained practical applications by applying and evaluating it on an unlabeled handwriting greeting card messages dataset.

Chapter 4

Automatic evaluation

In the previous stroke trajectory recovery evaluation system, we computed the distance between ground-truth stroke points and predicted stroke points. For this purpose, we require the ground truth strokes' coordinate information to compute the difference. However, adding the coordinates information is extra work and requires extensive labor. Moreover, the existing handwriting datasets have limited availability of stroke coordinates information. Hence, although the proposed system can be applied to handwriting datasets without stroke coordinates information, it is impossible to evaluate the quality of its predicted strokes using the existing methods. Therefore, we propose two measures for evaluating the quality of recovered stroke trajectories when ground truth stroke information is not given, namely image matching and readability.

4.1 Image matching

The LSTM predicts the stroke trajectory recovery point coordinates (x,y). We digitize the obtained strokes (X, Y) by plotting the points (x,y) and constructing a digital image from all the recovered stroke points (X, Y). As we compare the original and reconstructed image, the dimensions of the reconstructed image are the same as the original image. However, since the text is plotted with unit thickness, the thickness of the text in the input and reconstructed image differs. To overcome this issue, we propose dilating the reconstructed images so that the thickness of the text in the input and the reconstructed image are the same. We dilate for the kernel size ranging from 0 to 10 and select the kernel size that yields the least number of pixels in the absolute difference between the input and dilated images. Let I_{input} be the input handwritten text image, and let $I_{predict}$ denote the image reconstructed from the coordinates of stroke points predicted by LSTM. $I_{predict}$ is reconstructed by digitalizing the predicted stroke trajectory as described in Sec. 3.3.2. Hence all the words in $I_{predict}$ are one pixel thick.

Next, we dilate $I_{predict}$ with a dilation kernel $k \in (1, 10)$, and denote the dilated image with kernel k as $D(I_{predict}, k)$. We compare two digital images by computing their symmetric difference as

$$I_{Diff(k)} = |I_{input} - D(I_{predict}, k)|. \tag{4.1}$$

We define I_{Diff} as the image $I_{Diff(k)}$ with the minimum number of foreground pixels for $k \in (0, 10)$. This allows for estimating the thickness of the input text in the reconstructed image.

Next, we check the quality of the reconstructed image I_{Diff} by performing connected component analysis. Let C be the largest connected component in I_{Diff} image. The ratio of the number of foreground pixels in C to the total number of foreground pixels in I_{input} gives us the error in the stroke trajectory prediction, which is denoted as ϵ. This value can be used as a quantitative measure of the predicted strokes. Empirically, we observed that if the error ϵ is less than the threshold ($\mathcal{T} = 0.025$), then the quality of stroke trajectory recovery is good and vice versa.

The intuition of the proposed method is that the image I_{Diff} has small scattered connected components if stroke trajectory recovery is good. However, the $I_{Diff(k)}$ image has large and quite noticeable connected components if stroke trajectory recovery is of poor quality. The example of good and poor stroke trajectory recovery validated by the proposed method is shown in Fig. 4.2.

4.2 Readability

The second part of the automatic evaluation checks the preservation of the readability of the input text and the text from the recovered handwriting trajectory. To verify that the recovered stroke trajectory is read the same as the input handwriting word, we recognize the characters in both images. Let I_i and I_r be the two images for word from the input handwriting image and the one recovered from the proposed stroke trajectory recovery method, respectively. The text recognition on I_i and I_r gives us the string of characters for input word denoted by $\mathcal{W}_i = [w_1, w_2, ..., w_m]$ and the string characters from recovered stroke trajectory denoted by $\mathcal{W}_r = [w_1, w_2, ..., w_n]$, where m and n are the total character recognized in I_i and I_r. We utilize the pre-trained text recognition network [50] to compute $\mathcal{W}_i$ and $\mathcal{W}_r$. The difference between the two recognized strings is computed by the edit distance between the two strings. In our work, we compute the edit distance between two strings $\mathcal{W}_i$ and $\mathcal{W}_r$ with Levenshtein distance. Let the Levenshtein distance between two strings be d_{lev} and the number of characters in input string $\mathcal{W}_i$ is m. The readability error $\mathcal{R}$ is defined as d_{lev}/m, that is, the ratio of incorrect string matching to the total number of characters in the input string. Ideally, the Levenshtein distance (d_{lev}) and readability error ($\mathcal{R}$) are expected to be zero for good stroke trajectory recovery. Empirically, we noticed that the $\mathcal{R}$ less than $\mathcal{T} = 0.1$ results in satisfactory reconstruction, which we define as acceptable readability. This process is illustrated in Fig. 4.1.

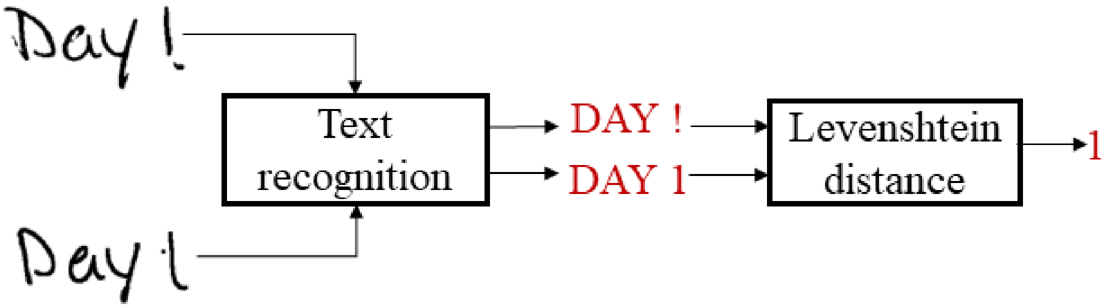


Figure 4.1: The readability module based on text recognition [50] utilized for our automatic evaluation. The top-left text is the input and the bottom-left text is reconstructed from predicted stroke trajectory recovery.

4.3 Experiments

We also applied the proposed automated evaluation method to the GHM dataset. In Fig. 4.2, we show the input handwritten messages, the dilated image, and connected components for I_{Diff}. The recovered image is dilated to match the width of the words in the input handwriting image as described in Sec. 4.1. According to the criteria defined in Sec. 4.1, the Fig. 4.2(top) example shows the *good* stroke trajectory recovery with small scattered connected components. Whereas, Fig. 4.2(bottom) shows the *poor* stroke trajectory recovery as there are larger connected components in the difference image I_{Diff}.

In Table 4.1, we listed the accuracy of the image matching and readability-based evaluation proposed in our work. The first column in Table 4.1 shows the percentage of documents with correctly recovered stroke trajectories according to our two proposed quality measures. According to the thresholds defined in Sec 4.1 and Sec. 4.2, the accuracy of stroke recovery from image matching and readability is 24.30% and 24.76%, respectively.

We manually verified the accuracy with user scoring.

If the threshold defined in image matching and readability evaluation classifies the image from the recovered stroke trajectory as satisfactory and the user also gives a satisfactory score to the recovered image, then a confidence score of 1 is assigned to that reconstruction. The average confidence score ($confidence$) for all the images is computed. The user scores the image in binary, scoring either 0 or 1. We applied this binary criterion to evaluate the robustness of the thresholds defined in Sec. 4.1 and Sec. 4.2. The accuracy of image matching and readability with the confidence score ($confidence$) are listed in Table 4.1. Our observation shows that the automatic evaluation based on image matching is a better

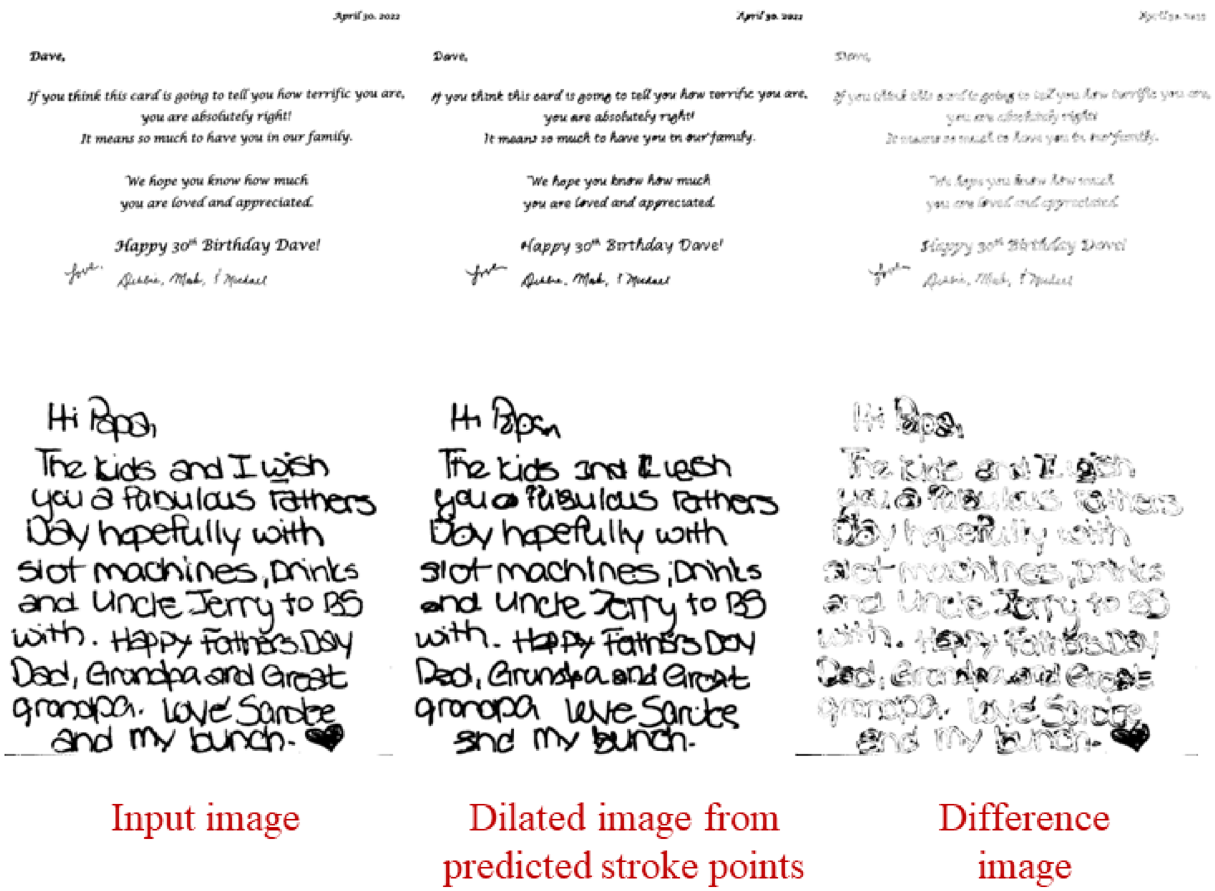


Figure 4.2: The input image, dilated image (after dilation is applied to the recovered image), and the difference of the input and dilated images.

evaluation measure than the readability evaluation. One of the reasons is that the text recognizer can correctly recognize words even if they are visually dissimilar to the input handwritten words.

Method	Accuracy	Confidence
Image matching	24.30%	74.0%
Readability evaluation	24.76%	53.8%

Table 4.1: The quantitative analysis of automatic evaluation on greeting cards handwritten messages [72].

Chapter 5

Local Feature Fusion with Character Shape Refinement for Handwriting Imitation

Handwriting image generation is a critical problem in document analysis as it facilitates handwriting recognition and writer's order estimation. The variety of human handwriting poses multiple challenges such as calligraphic style, slant, and font thickness imitation for the handwriting image generation process. The current handwriting generation networks use disentangled representation to learn the overall handwriting style. However, it is not sufficient to mimic the local character shapes of the letters. Moreover, the previous methods utilized the L1-loss function to match the ground truth and generated image, which overlooks to match the unique calligraphic shapes of the characters. In our proposed work, we introduced the loss function comprising pixel distribution matching in the ground truth and generated images. This loss function captures the local calligraphic shape of the letter. Moreover, we introduce the fusion of local attention features with global style features to improve the unique calligraphic style representation. For the proposed method, the experimental analysis gives 12.9% us better results for image generation for pixel distribution matching.

Although there are recent methods for generating photorealistic images for natural scenes, generating handwriting images is still an open research problem. In contrast to the natural scenes, handwriting images of varying lengths depend on the number of letters in each word. Secondly, the handwriting images may contain arbitrary textual content containing words out-of-vocabulary. Thirdly, humans have varying handwriting styles with different handwriting strokes, cursive ligatures, and font sizes. Therefore, no fixed templates could be followed for handwriting image generation.

Recently, several studies have proposed GAN architecture for handwriting generation, with most methods offering limited advantages for handwriting image generation. First network [4] proposes GANs for synthesizing handwritten text images utilizing the entire word embedding for style representation. However, this method can low visual qualities for OOV words. ScrabbleGAN [22] can synthesize arbitrarily long handwritten texts by concatenating

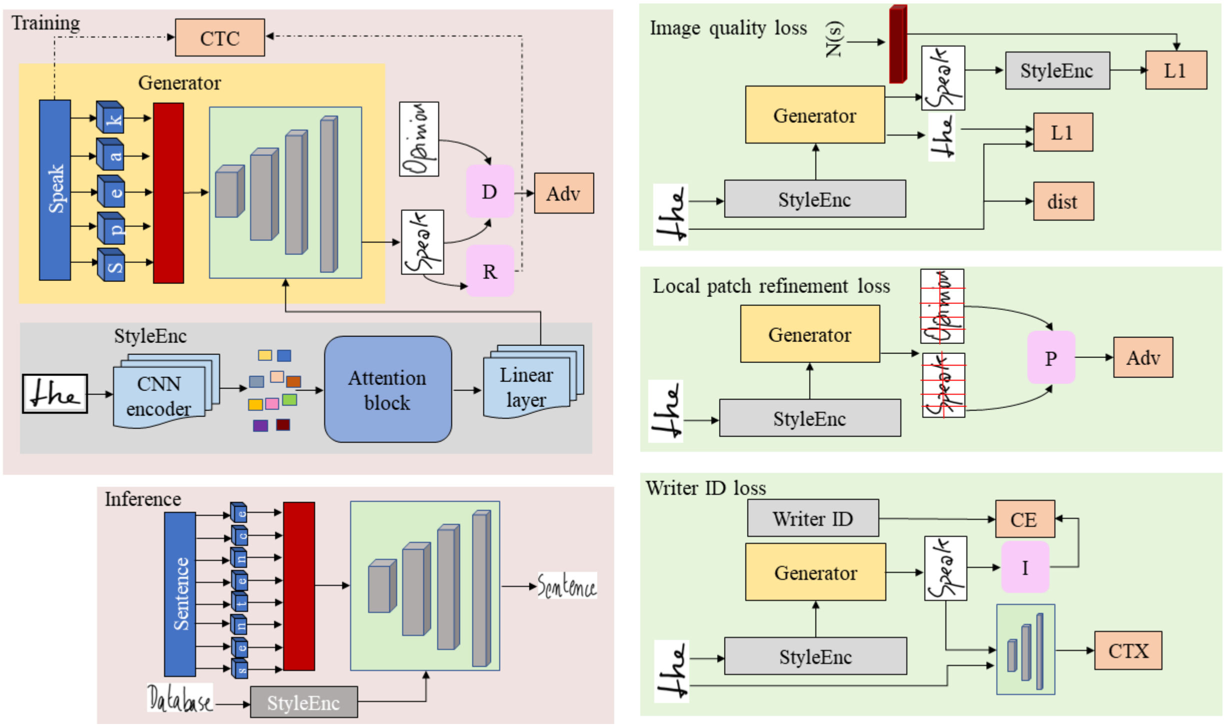


Figure 5.1: The overall network architecture

all the letter tokens, but it cannot imitate calligraphic styles of reference samples. [43] proposed a few-shot style-conditioned handwritten word generation. However, it is limited to synthesizing short words rather than long texts.

Moreover, Bhunia [10] proposed a transformer-based architecture for handwriting image generation. .[10] takes a long to train with the help of recognition and writer identification network and cannot perform style imitation.

Recently, [25] is proposed to utilize GAN based framework for handwriting image generation. It offers the advantage over the previous method, which could be trained using a single-word image as a sample. On the other hand, the previous methods require at least 15 sample images to train the GAN architecture. The generated image quality is still unsuitable for unique character details, and images are also blurred in overall quality. To offer the image quality improvement to [25], a [26] proposed to include a patch discriminator to the GAN architecture. These patch discriminator are fed by local patches of the image and trains to improve the blur in the generated images.

The handwriting image generation methods mentioned above [10, 22, 25, 26, 43] are promising for producing photorealistic images but cannot imitate the unique character styles in sample images. The style encoder used in the previous methods utilizes a residual convolutional network for style representation learning from the entire image.

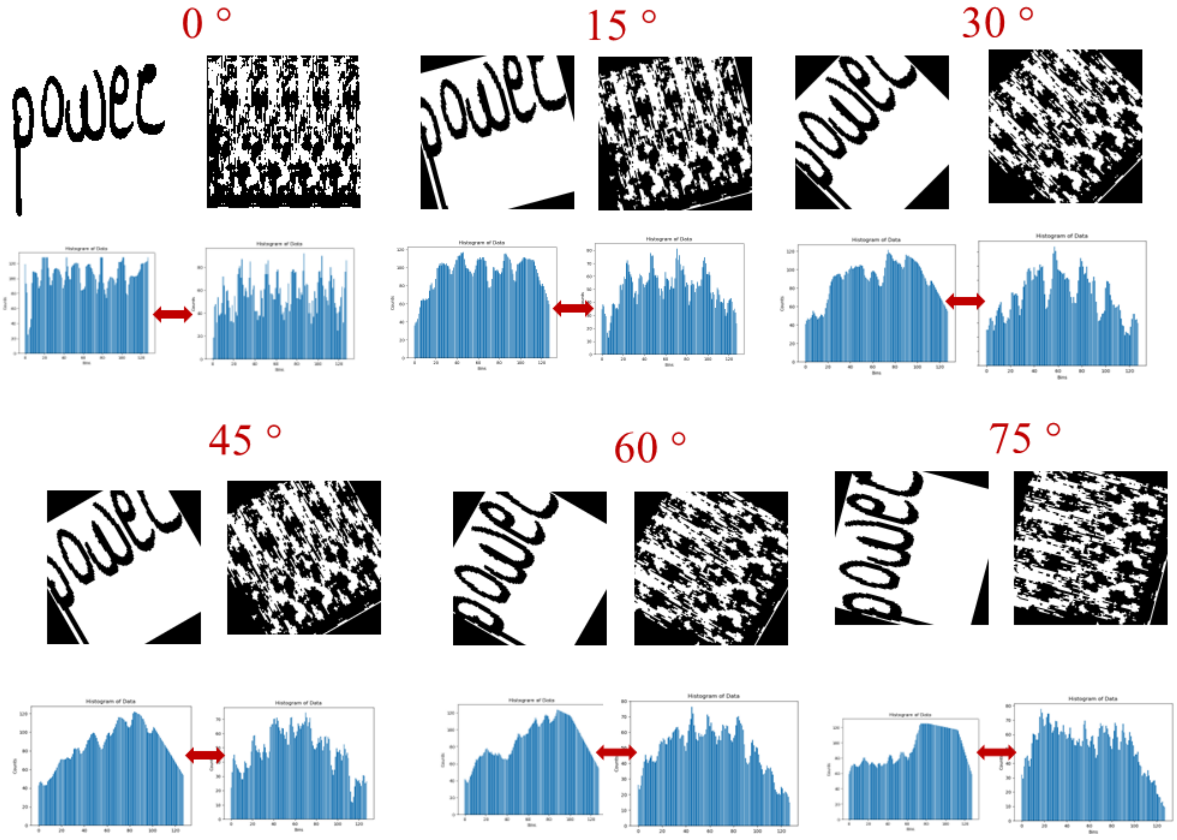


Figure 5.2: The generated and ground-truth image and their respective histograms

5.0.1 Problem statement

Every writer draws individual characters in a specific character shape, and the overall handwriting styles, such as slant, character spacing, and linkage in any handwriting style, are different. In our work, we propose a solution for handwriting imitation conditioning on any calligraphic style with particular attention to character shape $\hat{x} = G(y)|E_{(w,c)}$. The proposed system learns the style embedding $E_s(w, c)$ from word images to capture the local character shapes. Moreover, in our work, we introduce to incur loss based on the shape of the characters. The shape of the characters in words is estimated by the distribution of foreground pixels in the generated image. As a result, our handwriting imitation model generates characters in words more similar to the sample handwriting style.

5.1 Method

5.1.1 Network architecture

Textual content embedding

There are mainly two ways to encode the features from the textual content, first is to extract features from an entire string of text and convert it into a fixed-length feature representation. The explicit way is to encode each character embedding c and concatenate these character tokens into a feature representation. This improves the generalizability of the generative model as highlighted in [26]. In this way, our generative model is not restricted to learning the character-level handwriting imitation limited to the words in the training corpus.

For the text content $y = [y_1, y_2, ..., y_l]$ with l-characters, the variable length feature map is computed by concatenating features from each character as $\mathcal{F} = [f_y, f_y, ..., f_y]$. This feature vector is provided as textual content embedding into a generative network to generate the image for the sample handwriting style.

Adversarial generative model

The textual content features $\mathcal{F}$ are fed into generative model $\mathcal{G}$ along with character-attention style embedding $\mathcal{S}$. The generator network is fully convolutional to ensure the variable length for image generation based on the number of characters in the textual content image. Our style feature network is specifically designed to focus attention on the characters and calligraphic style of handwriting simultaneously to encode the character shape and calligraphic style (text slant, font thickness, and ligature) into a unified character-attention style feature embedding. Our generator network $\mathcal{G}$ aims to mimic the character's unique shape as well as the overall handwriting style.

Global and local patch discriminators

In our work, we leverage two discriminators, one for the entire generated image (global discriminator) and the second one for the cropped patches from the generated image (local discriminator). The global discriminator tried to classify a generated image as real/fake. However, the local discriminator attempts to grade the cropped patches from images either belonging to the generated image or to the training dataset. The global and local discriminators to refine the generated patches of handwriting are inspired by [26].

Supporting networks

In our framework, we used a pre-trained writer identification network to classify the handwriting images into their respective writers and it guided the generator to imitate the style of

Method	IS	FID	KID	PSNR	MSSIM	WER	ADP
ScrabbleGAN	1.3268	26.7758	2.9479	11.2562	0.1950	0.0740	-
ST-GAN	1.2443	33.9069	3.1314	12.0345	0.1845	0.1968	-
GANwriting	1.3267	20.5539	1.3927	10.8045	0.2038	0.2143	-
HWT	1.3620	19.6938	1.8003	10.7518	0.2319	0.1032	-
HiGAN	1.3298	18.3095	1.6688	11.7609	0.2459	**0.0085**	-
HiGAN+	1.4059	**5.9510**	**0.3709**	12.3391	0.3322	0.0186	0.05917
λ_{ADP}	**1.4078**	6.2809	0.3834	**12.399**	**0.3375**	0.0223	**0.0515**

Table 5.1: The quantitative comparison of our method with state of the art methods for the handwriting imitation

	Method	IS	FID	KID	PSNR	MSSIM	WER	ADP
A	$\lambda_{adv} + \lambda_{ctc}$	1.2975	24.3985	2.2173	11.6510	0.2078	0.0243	-
B	A + $\lambda_{recn} + \lambda_{kl}$	1.3356	25.2118	2.0649	11.3832	0.2452	**0.0080**	-
C	B + $\lambda_{id} + \lambda_{style}$	1.3298	18.3095	1.6688	11.7609	0.2459	0.0085	-
D	C + λ_{ctx}	1.3694	10.6346	0.7752	11.9286	0.2981	0.0085	-
E	D + λ_{patch}	1.4059	**5.9510**	**0.3709**	12.3391	0.3322	0.0186	0.05917
F	λ_{ADP}	1.3782	6.8078	0.4228	11.6530	0.2766	0.0231	0.1293
G	$\lambda_{recn} + \lambda_{ADP}$	**1.4078**	6.2809	0.3834	**12.3990**	**0.3375**	0.0223	**0.0515**
H	G + local attn.	1.3682	7.5499	0.4521	11.9364	0.2839	0.0361	0.0960
J	H + λ_{ADP}	1.3537	10.8053	0.7960	11.8857	0.2705	0.0809	0.0860

Table 5.2: The ablation study for different loss functions for handwriting imitation

the handwriting based on writer identification. The other pre-trained component in our system is the word recognition network. The word recognition network ensures the generation of readable text for the given textual content and style embedding.

character-attention style features

In our work, we propose a character patch self and cross attention to learn the style embedding representation from sample style images. We crop the character patches from word images by a character detection network described in Sec. 6.1.2. The character detection network is trained without character bounding box annotations in a weakly supervised way with only available character boxes for synthetic scene text datasets. Character detection provides us with a reasonable estimation of the location of characters with unique shapes.

The attention scores of all the characters with themselves as well as with the entire word are computed. For this purpose, we design a self and cross-attention encoder network. The number of characters in each word is different so we introduce the blank image padding to ensure the number of characters passes to the encoder remains the same despite the different number of characters in each word. An attention mask is defined on the basis of the number of characters in each word. we introduce an encoder attention mask to avoid the contribution of attention scores from padded images. The block diagram of the proposed architecture is shown in Fig. 5.1.

5.1.2 Loss functions

shape matching loss

The previous handwriting imitation work [26] suggests the Euclidean L1-distance between generated $\hat{x}$ and ground-truth x image to penalize the difference between two images. The Euclidean L1 distance works well for global image quality since it takes the average of the difference of pixels in two images. However, it does not seem to be sensitive to the calligraphic style of individual characters in handwriting. Therefore, in our work, we introduce to match the distribution of pixels in generated $\hat{x}$ and ground-truth x image in different orientations as follows:

$$\mathcal{L}_p(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{P} \sum_{p=1}^{P} \mathcal{L}_{1d} \left(\phi_p(\boldsymbol{x}), \phi_p(\hat{\boldsymbol{x}}) \right) \tag{5.1}$$

where, $\phi_p(\boldsymbol{x})$ and $\phi_p(\hat{\boldsymbol{x}})$ are the distribution of foreground pixels in p^{th} orientation. To the best of our knowledge, our work is the first to introduce distribution matching loss for handwriting imitation.

Reconstruction loss

Global and local adversarial loss

In our network, we employed global and local adversarial losses. The generative network G is trained both for global and local adversarial losses. The global adversarial loss $\mathcal{L}_{G_{adv}}$ distinguishes between the fake images generated by the generator G given the textual content and style representation of the sample image and real image sampled from the training datasets for the same writer.

Let's suppose, for the textual content $T \in C$ and style representation s, G generates a fake image $\hat{x} = G(T|s)$, where the style features s is extracted from sample image w and set of character patches c given as $s = E_{(w,c)}$.

$$\mathcal{L}_{G_{adv}} = \mathbb{E}_x[\log D(x)] + \mathbb{E}_{T,s}[\log(1 - D(G(T|s)))] \tag{5.2}$$

The general adversarial loss is used to train the generative adversarial network. However, to produce a clear handwriting image with minimum local blur, we also employed an adversarial loss on the handwriting image $\hat{x}$ for local patches cropped from the generated image. Similar to classifying the image as real or fake, we employed a patch discriminator

P which classifies the patches of images as cropped from the real or fake image. For real $(\psi_i^x \mid i = 1 \cdots M)$ and fake $(\psi_i^{\hat{x}} \mid i = 1 \cdots M)$ patches, the patch adversarial loss inspired by [26] is given as:

$$\mathcal{L}_{\text{patch}} = \frac{1}{M} \sum_{i=1}^{M} \left\{ \mathbb{E}_x \left[\log P\left(\psi_i^x\right) \right] + \mathbb{E}_{\hat{x}} \left[\log\left(1 - P\left(\psi_i^{\hat{x}}\right)\right) \right] \right\}, \tag{5.3}$$

where M is the total number of patches cropped from x and $\hat{x}$. The examples of ground truth and generated image with their corresponding histograms are shown in Fig. 6.5

5.2 Experimentation

5.2.1 Dataset

We will use the IAMonline dataset [54]. The dataset [2] consists of 11,192 lines of text with 58,844 words. DeepWriting provides characters for individual writer styles. There are 467 writers, out of which 372 writers are used for training the network and the rest are used to evaluate the network's performance.

5.2.2 Results

We compared our proposed method with previous state-of-the-art methods [10, 22, 25, 26, 43]. The previous methods are utilizing GAN architecture with style learning extracted from entire image. The quantitative results of the proposed method are listed in Table 5.1. Here, we can see that the proposed method λ_{ADP} is performing better than the state-of-the-art method for image generation evaluation in the case of image-similar learning metrics such as PNSR and MSSIM. We also show an ablation study of the proposed method in Table 5.2.

There is an interesting observation that the inception feature-based evaluation metric is better for state-of-the-art methods. Therefore, the distance between inception features (FID, KID, IS) is not reflecting of image quality similarity between ground truth and generated image. In Fig. 5.3, we also show some qualitative results for the proposed method compared to the state-of-the-art method. We can see that the bottom-right block (local attention with projection loss)shows better handwriting imitation compared to the rest of the methods.

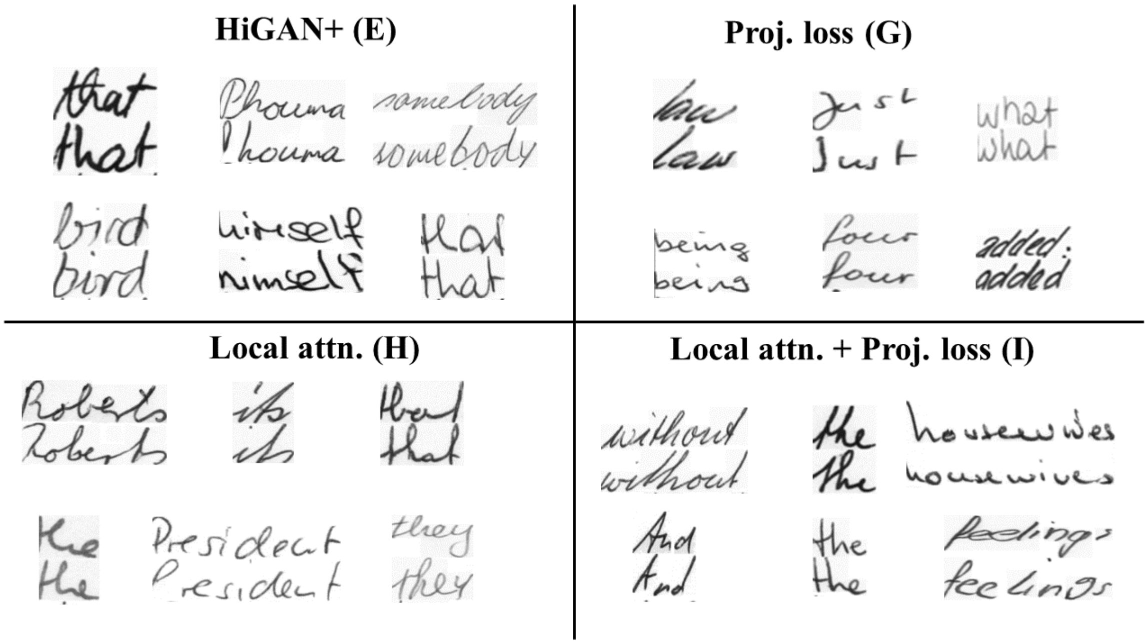


Figure 5.3: The quantitative comparison of previous [26] and proposed method

Chapter 6

MS-CAP: Probabilistic Diffusion Model Conditioned on Multi-Scale ChAracter Pairs for Style in Handwriting Stroke's Generation

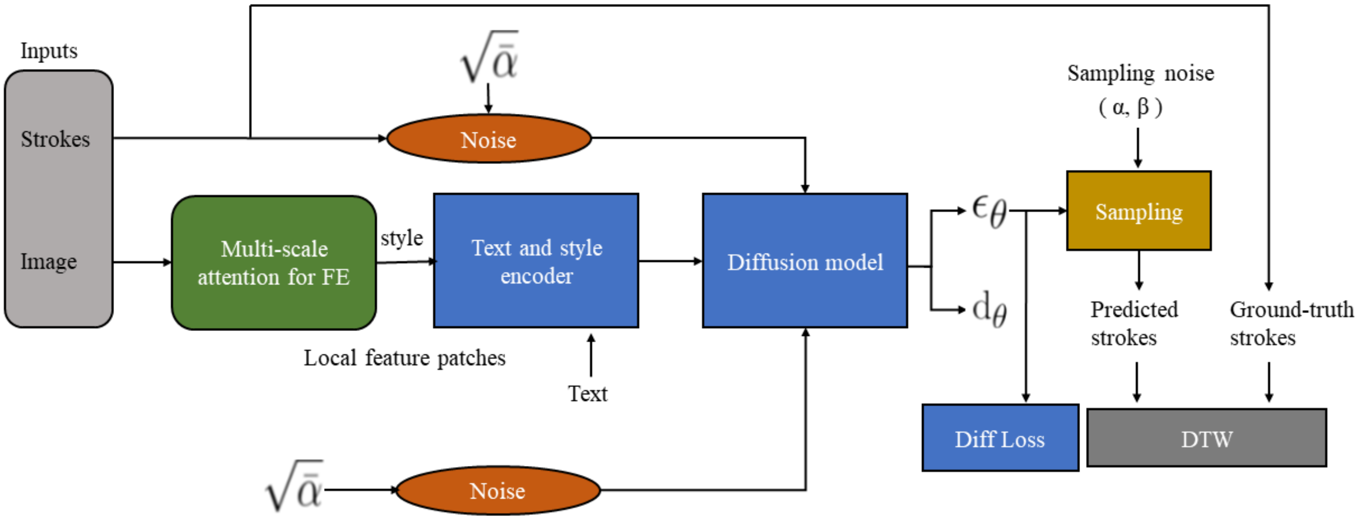


Figure 6.1: The overall block diagram of diffusion model with multiscale attention features for the style image. It also includes Dynamic Time Warping (DTW) loss for stroke sequence matching to train the diffusion model.)

6.1 Method

The proposed method has four main components: multi-scale attention for style feature extraction, text-style encoder, diffusion model, and loss function. A high-level block diagram of the proposed method is shown in Fig. 6.1. We describe the multi-scale attention (MS) features for handwriting images in Sec. 6.1.1. To include the additional style features that signify the connectivity of character pairs, we introduce multi-scale character pair ($MSCAP$)

features for text-style encoder as elaborated in Sec. 6.1.2. The overview of the diffusion model for handwriting stroke generation is described in Sec. 6.1.3. Finally, in Sec. 6.1.4, we discuss the addition of Dynamic Time Warping (DTW) loss as an auxiliary loss to train the diffusion model.

6.1.1 Multi-scale attention features

Generally, multi-head attention networks process images at a fixed resolution [89, 31]. However, handwriting images may constitute different font sizes, word spacing, and handwriting styles. To extract the style features at different granularity levels in the calligraphic style, we propose to compute multi-head attention features at multiple scales. Intuitively, multi-scale attention focuses on details in the original resolution images and on more global calligraphic style in the down-sampled variants.

The previous methods for handwriting imitation (via stroke generation) [53, 45] also compute the features from the style template images. However, the style features in the earlier methods are not distinctive for different handwriting styles since they are calculated from pre-trained networks [53] trained on the natural images. These networks cannot extract discriminative local features for character shapes and overall style. We propose the local features from the handwriting-style template image at multiple scales to learn the character shape with global handwriting styles. For this purpose, we consider images at three scales. One at the original image resolution and two down-sampled variants with their aspect ratio preserved (ARP). As we are using three scales, we divide the respective images (original image and two variants) into equal-sized patches at each scale. The details of the multi-scale attention network are as follows.

Patch embedding

In handwriting images, feature representation from the character shape and global style plays a vital role in capturing the overall style for handwriting imitation. The proposed multi-scale handwriting style representation helps to capture the global and local style information. Patches from different scales enable the attention mechanism to aggregate information across multiple scales and spatial locations.

The input for our multi-scale attention comprises the full-size image with height H, width W, channel C, and two ARP resized variants using a Gaussian kernel. The downsampled variants have height h_k, width w_k, channel C, where k = [1, 2] since we are using two resized variants. The input image is a 128x1024 dimensional grayscale image. The two variants are down-sampled at 96x768 and 64x512 resolution. In our work, the feature representation from down-sampled images improves the quality of the feature's representation and makes them independent of the quality of the input handwriting image.

Square patches of size $P \times P$ are extracted from each image in the multi-scale representation. We pad the image with zeros if the width or height is not multiples of P. A 5-layer ResNet [37] learns the D dimensional patch embedding of each patch with a fully connected layer of size D.

The patch embedding module is intended to compute the embedding of each patch, assigning a unique embedding to every patch across different scales. Consequently, patches that look visually similar and are located in the same position may have different embeddings in each scale, despite their visual similarity. Yet, the ideal characteristic for positional embedding is that spatially proximate patches should share the same positional embedding, regardless of whether they belong to different scales.

To satisfy this property, we describe a spatial embedding in the next section, which ensures that the spatially close patches in different scales have the same spatial embedding.

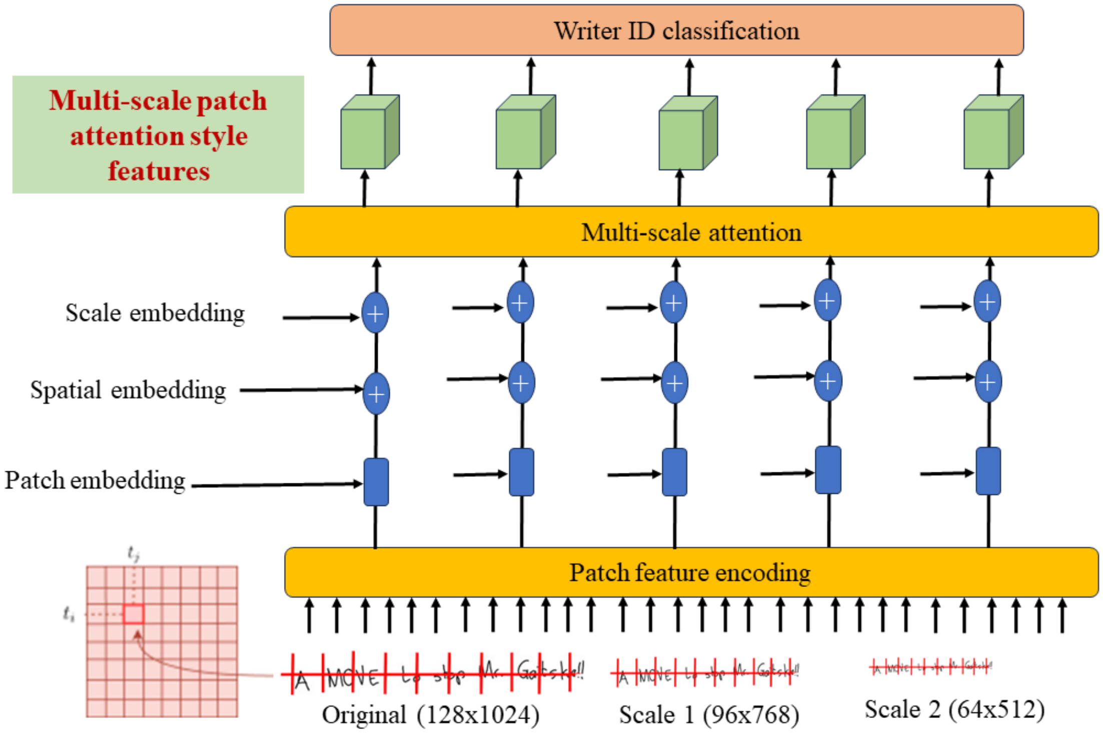


Figure 6.2: The architecture for multi-scale learning with attention via writer identification.

Spatial embedding

As we have mentioned before, we leverage patches from different scales for better style features for handwritten images. It also imposes an additional constraint on the positional embedding. The patches from different scales corresponding to the same image portions should have the same spatial embedding. The spatial embedding is required to follow the set of requirements such as 1) effectively encode the 2D spatial position of each patch to 1D sequence; 2) spatially close patches at different scales should have close spatial embedding; 3) efficient for computing the multiscale attention. On the other hand, traditional positional embedding assigns different patch embedding to each patch and is not able to align the spatially close patches from different scales. Based on that, we utilize hash-based 2D spatial embedding (HSE). The patch at the location (row i, column j) is hashed to the corresponding element in a $G_h \times G_w$ grid, where each element in the grid is a D-dimensional embedding. *HSE* is defined by a learnable matrix $T \in R^{G_h \times G_w \times D}$. The input with resolution $H \times W$ is partitioned into $\frac{H}{P} \times \frac{W}{P}$ patches. For the patch at position (i, j), its spatial embedding is defined by the element at position $(t_i,\, t_j)$ in T where

$$t_i = \frac{i \times G_h}{H/P}, t_j = \frac{j \times G_w}{W/P} \tag{6.1}$$

The patch located at row i column j of the image is hashed to the corresponding element $(t_i,\, t_j)$ of matrix T. The D-dimensional spatial embedding T_{t_i, t_j} is added to the patch embedding element-wisely as shown in Fig 6.2.

To ensure alignment of patches across different scales, we map the patch locations from all scales onto a common grid T.

As a result, patches located closely in the image but from different scales are mapped to spatially close embeddings in T, since i and H, as well as j and W, change proportionally to the resizing factors. The hash spatial embedding is inspired by [44].

We selected the appropriate grid size through experimentation. As we know, handwriting sentence has a longer length than their height, so using the same grid size for width and height dimensions is not an appropriate choice. Therefore, we propose using the smaller grid size for height compared to its width. For the IAM-online [54] dataset, we used $[(G_h, G_w)\ ]=$ [4x32] grid size as an appropriate choice, where G_h is a grid size for height and G_w is a grid size of the width. Smaller G_w may result in a lot of collision between patches, making the model unable to distinguish spatially close patches. Larger G_w requires large memory and may need more diverse resolutions to train. The smaller size of G_h might result in overlapping patches however the larger values of G_h would not be able to capture the local feature across the height dimension of the handwriting. The block diagram of multi-scale attention with patch and spatial embedding is shown in Fig. 6.2.

Scale embedding

The spatial embedding satisfies the condition to assign the same embedding to spatially close patches in different scales. However, it does not distinguish information coming from different scales. So, we define another embedding called scale embedding to help the attention model to distinguish information coming from different scales effectively.

Online writer ID	
Methhod	%acc
Baseline [53]	4.90
Multi-scale [4, 32]	**32.35**
Offline writer ID	
Method	%acc
Baseline [37]	87.07
Multi-scale [4,32]	**95.60**

Table 6.1: Classification accuracy for writer identification for online and offline handwriting. The multi-scale attention outperforms the baseline for both online and offline by a large margin.

We define scale embedding as a learnable embedding Q $\in R^{(K+1)D}$ for the input image and two downsampled variants. Following the spatial embedding, the first element $Q_0 \in R^D$ is added element-wise to all the D-dimensional patch embeddings from the original image resolution. $Q_k \in R^D$ k = 1, 2 are also added element-wisely to all the patch embeddings from the downsampled variants. The sum of patch embedding, spatial embedding, and scale embedding are fed into a multi-head attention network. We train our multi-scale attention network for writer identification from handwriting images.

Table 6.1 shows the accuracy of writer identification for online and offline handwriting images. We compare the proposed methods with baseline mobile net [53] for online handwriting and residual network [37] for offline handwriting. We can see that the multi-scale features outperform the baseline for both online and offline by a large margin. Baseline methods are CNN networks Which have the disadvantage of extracting features at fixed image size. However, we extracted multi-scale features which improve the overall writer identification accuracy.

The local patch features of dimensions 77x384 are extracted from the multi-scale attention network before the classification layer. The 77 local patches each of dimensions 384 embeds the local and global style information of handwriting style. These local patches with rich style information are used to train the diffusion model in Sec 6.1.3. To the best of our knowledge, the representation capabilities of multi-scale patch embedding, spatial embedding, and scale embedding have not been explored before to represent handwritten images' local character and global style features. Our experimentation in Sec. 6.2.3 validates the effectiveness of these features for handwriting stroke generation.

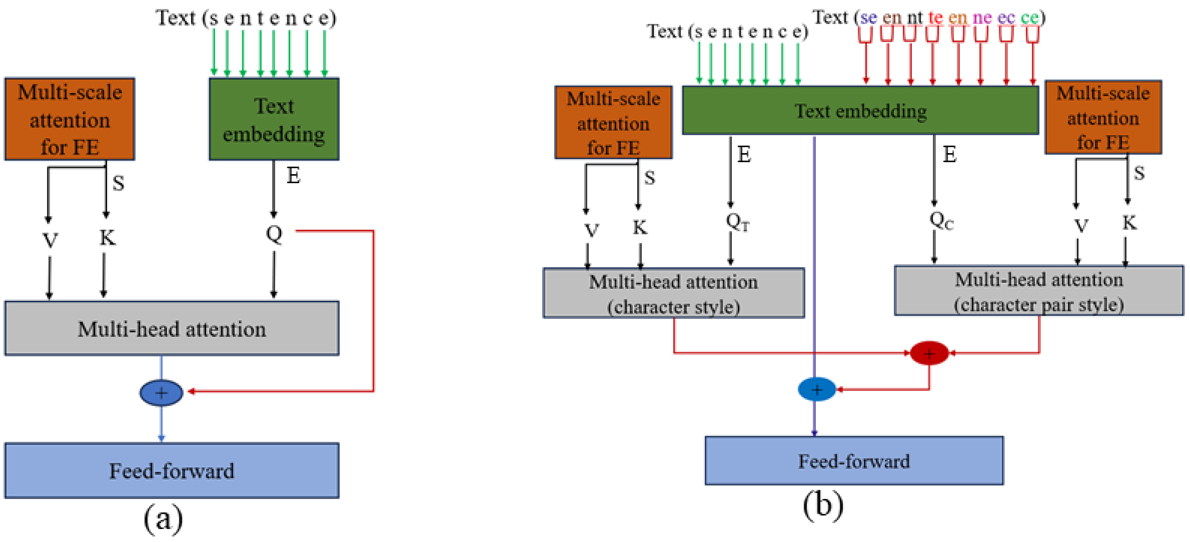

Text (s e n t e n c e)
Multi-scale attention for FE
Text embedding
S
E
V
K
Q
Multi-head attention
Feed-forward
(a)
Text (s e n t e n c e)
Text (se en nt te en ne ec ce)
Multi-scale attention for FE
Text embedding
Multi-scale attention for FE
S
E
E
S
V
K
Q_T
Q_C
V
K
Multi-head attention (character style)
Multi-head attention (character pair style)
Feed-forward
(b)

Figure 6.3: (a) Block diagram of a text-style encoder for stroke diffusion [53], (b) Block diagram of a text-style encoder for modifying style features with character style as well as the connection between characters.

6.1.2 MSCAP text-style encoder

The multi-scale (MS) features introduced in the last section are effective in encoding a local character shape as well as the global handwriting style. For style S conditioning, we first extract N local patch features of each k dimensions from the handwriting image. The text embedding module embeds each character from text string T into K-dimensional embedding E. The attention $attn$ mechanism computes the attention of each text character embedding to each patch of the local style features. The block diagram to show the attention between style features and text sequence is shown in Fig. 6.3(a). The attention output is then added to the text sequence embeddings as E + attn(S, E) before passing through a feedforward network, which learns the compact style-text encoding.

Fig. 6.3(a) shows the compact text-style encoding based on local style patches and text embedding. It encodes each character into the writer style. For instance, each character in text (*sentence*) is utilized separately to attend to style features. However, it lacks the style representation for character pairs such as *"se"*, *"en"*, *"nt"*,... *"ce"*. In our work, we propose a style-text attention mechanism with the aim of enhancing the writer's style by incorporating the calligraphic element between pairs of characters.

Here, we get two sets of style features: one is character attention attn(S, T), and the other is character pair P attention attn(P, T). The character attention computes the local character shape of the writer's style. However, the character pair's attention captures the connection between the characters. In our proposed work, we attend the character pairs such as *"se"*, *"en"*, *"nt"*,... *"ce"* to the style features as well. Finally, we sum up three features, namely character attention, character pair attention, and text embedding as E + attn(S, E)

+ attn(P, E) as shown in Fig. 6.3(b).

We condition the diffusion model on the proposed text-style features for handwriting stroke generation. The style is extracted from the handwritten image template, and text content is the given arbitrary text that we intend to generate in the same style as the style template. This method of attention between style features and text sequence is effective in conditioning the diffusion model for stroke prediction. Our feature extraction method surpasses the baseline [39] for handwriting style features proposed in [53].

In Section 6.2, we demonstrate the effectiveness of our proposed text-style features for handwriting imitation via stroke prediction. To the best of our knowledge, this is the first study to explore the multi-scale handwriting style features as well as the character pair attention to condition the diffusion model for handwriting stroke generation.

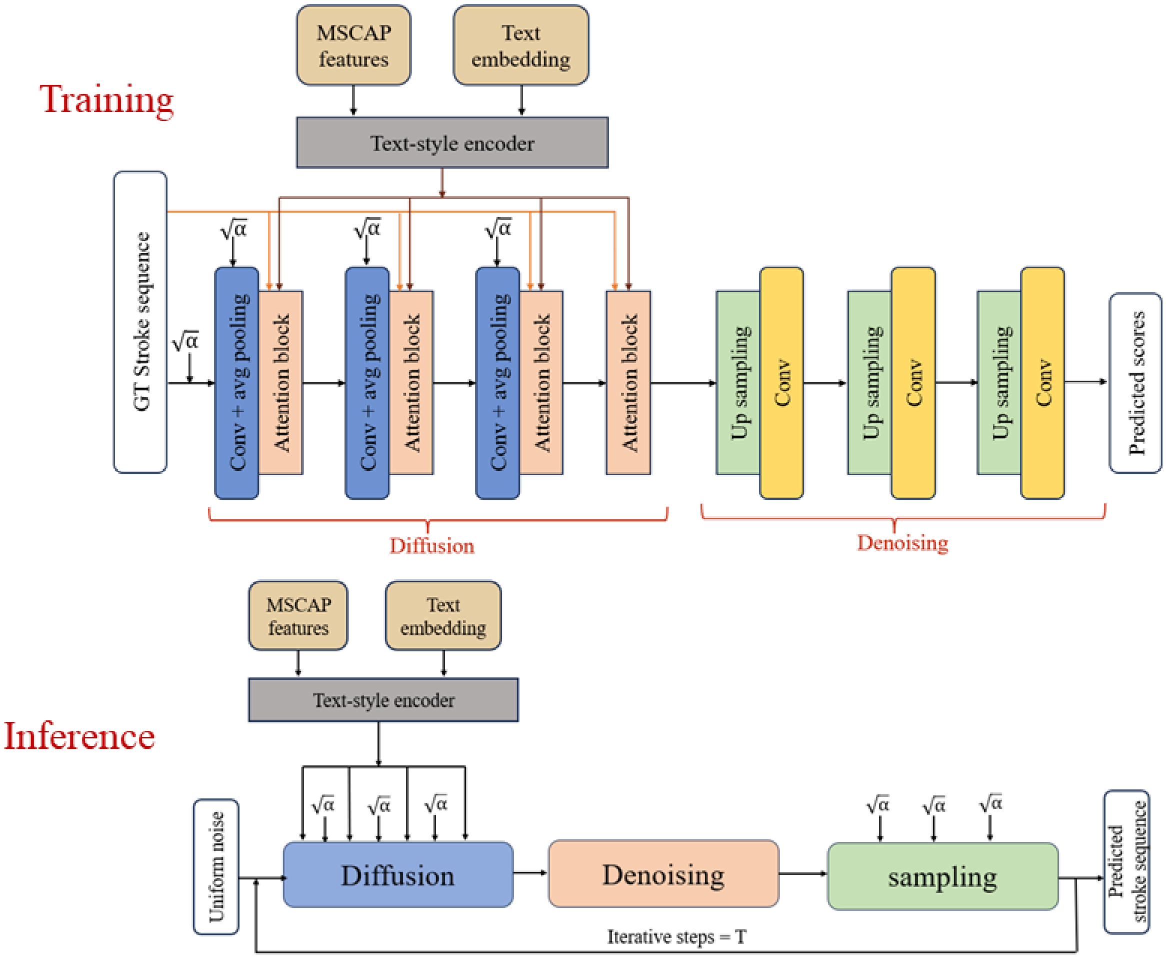


Figure 6.4: The internal layer-wise architecture of diffusion model for strokes prediction from handwriting images for training and inference phase.

6.1.3 Stroke diffusion Model

Our diffusion model is conditioned on text-style embedding from Sec. 6.1.2. It comprises a set of convolutional layers with attention blocks. We iteratively add the Gaussian noise to the ground truth stroke sequence. In general, the diffusion model employs Markov chains to add noise and disrupt the structure of input data, this step is called the diffusion process. In the reverse process, the model then learns to reverse the diffusion process and tries to reconstruct the original data, this process is called the denoising process [74].

In the diffusion process, a sample x_0 is initially drawn from a distribution $x_0 \approx q(x_0)$ corresponding to the original data with the same dimensionality as x_0. Adding a Gaussian noise to this distribution produces a latent variable y_1; noise is again added to y_1, giving latent variable y_2, and so on, We repeat this process for T steps to form a series of latent variables y_1, y_2,. . . ,y_T during the diffusion process. In training a diffusion model, we define a series of hyper-parameters β_1, β_2, ..., β_T which collectively form a noise scheduling for adding the Gaussian noise that perturbs the input

Let $q(y_0)$ be the data distribution, and let $y_1, ..., y_T$ be a series of T latent variables with the same dimensionality as y_0. Mathematically, the diffusion process $q(y_{1:T}|y_0)$ is defined as a fixed Markov chain where Gaussian noise is added at each iteration based on a fixed noise schedule $\beta_1, ..., \beta_T$ as stated below:

$$q\left(\mathrm{y}_{1:T} \mid \mathrm{y}_0\right)=\prod_{t=1}^T q\left(\mathrm{y}_t \mid \mathrm{y}_{t-1}\right) \tag{6.2}$$

$$q\left(\mathrm{y}_t \mid \mathrm{y}_{t-1}\right)=\mathcal{N}\left(\mathrm{y}_t ; \sqrt{1-\beta_t} \mathrm{y}_{t-1}, \beta_t \boldsymbol{I}\right) \tag{6.3}$$

In the diffusion process, we keep on adding the noise so that the final ground-truth strokes become a pure Gaussian noise sample with no information about the original stroke sequence.

In the reverse (denoising) phase, a decoder network learns to gradually remove the noise from the sampled distribution until it ends up with the original stroke sequence. The denoising process is defined as a Markov chain parameterized by θ:

$$p\left(\mathrm{y}_T\right)=\mathcal{N}\left(\mathrm{y}_T ; 0, \boldsymbol{I}\right), \quad p_\theta\left(\mathrm{y}_{0:T}\right)=p\left(\mathrm{y}_T\right) \prod_{t=1}^T p_\theta\left(\mathrm{y}_t \mid \mathrm{y}_{t-1}\right) \tag{6.4}$$

where $p_\theta\left(\mathrm{y}_{t-1} \mid \mathrm{y}_t\right)$ intends to reverse the effect of the noise adding process $q\left(\mathrm{y}_t \mid \mathrm{y}_{t-1}\right)$ as follows:

$$p_\theta\left(\mathrm{y}_{t-1} \mid \mathrm{y}_t\right)=\mathcal{N}\left(\mathrm{y}_{t-1} ; \mu_\theta\left(\mathrm{y}_t, t\right), \Sigma_\theta\left(\mathrm{y}_t, t\right)\right) \tag{6.5}$$

where $\Sigma_\theta\left(\mathrm{y}_t, t\right) = \sigma_t^2 \mathrm{I}$ and σ_t^2 is a constant related to β_t.

Note that our diffusion model consists of consecutive convolution layers and attention blocks as shown in Fig. 6.4. The attention block computes the attention between ground-truth stroke sequence and text-style features. For training, the attention blocks condition the diffusion model on text-style features and noisy ground truth stroke sequence. The decoder part also includes a convolutional layer with upsampling layers which is designed to predict the noise scores. We consider a diffusion model [65] as a score-based generative model, where instead of learning to model the energy function itself from latent distribution, we learn the score of the energy-based model as a neural network.

The success of the conditional diffusion model for image generation makes it a suitable technique for generating handwriting images conditioned on the writer's style for handwriting imitation. The image generation in the diffusion model is learned by iteratively adding small amounts of noise to an image and changing it into a random image during training. The model learns to reverse this process, generating realistic images by removing noise. The emerging image generation models are based on diffusion models [65, 19, 15]. However, image generation is a computationally expensive process. Therefore, our work proposes to generate the stroke sequences using a diffusion model conditioned on the writer's style and textual content. It could be trained in a reasonable amount of time (see Table 6.5). It can also predict additional temporal information in the form of stroke sequence for the writer's handwriting which is not available for image generation [25, 26, 57, 10, 59].

Inference

During sampling, diffusion models iteratively remove the noise added in the diffusion process, by sampling y_{t1} for t = T, ... , 1. The stroke sequence y_{t1} at time step *T-1* is computed with the equation below.

$$\mathrm{y}_{t-1}=\frac{1}{\sqrt{a_t}}\left(\mathrm{y}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(\mathrm{y}_t,t\right)\right)+\sigma_t z \tag{6.6}$$

where z $\approx$ N(0; I) and α_t is a constant related to β_t. For our experiments, we used $\alpha_t^2=\beta_t$.

In our diffusion model, we sample uniform noise to provide an input stroke sequence to predict the stroke sequence in the desired style for given textual content. The diffusion model is provided with the text-style embedding of the desired style from the image and textual content from a text string. In this way, we do not need the ground truth strokes in the inference phase and the learned diffuion model effectively generates strokes from noise given text-style embedding. During inference, we perform diffusion and denoising process with the addition of sampling process as shown in Fig. 6.4. Our diffusion model design facilitates us to generate strokes for any arbitrary calligraphic style given any arbitrary text content.

6.1.4 Loss function

Diffusion loss

The output of our handwriting stroke generation x is composed of a sequence of N vectors $x_1 \ldots x_N$.

Each vector in the sequence x_i is composed of a real-valued pair, which represents the pen offset from the previous stroke in the x and y direction along with a binary entry that has a value of 0 if the pen was writing the stroke and 1 otherwise.

Each handwritten sequence is associated with a discrete character sequence C describing the text. Each sequence is also associated with an offline image containing the writer's style information, denoted by S. Since We cannot parameterize the binary variable representing whether the stroke was drawn by a Gaussian distribution as we did for the real-valued pen strokes. Instead, we parameterize it with a Bernoulli distribution. For this purpose, we split each data point x_i into two sequences y_i and d_i of equal length, with y_i representing the real valued pen strokes, and d_i representing whether the stroke was drawn. At each step t, our model d_θ returns an estimate $\hat{d_i}$ of whether the pen was down.

$$L_{\text{stroke}}(\theta) = \left\|\epsilon - \epsilon_\theta\left(\mathrm{y}_t, \mathrm{c}, \mathrm{s}, \sqrt{\bar{\alpha}}\right)\right\|_2^2 \tag{6.7}$$

$$L_{\text{drawn}}(\theta) = -\mathrm{d}_0 \log\left(\hat{\mathrm{d}}_0\right) - (1 - \mathrm{d}_0) \log\left(1 - \hat{\mathrm{d}}_0\right) \tag{6.8}$$

Diffusion models are generally trained with diffusion loss, as mentioned above. However, recent attempts on image generation [32, 90, 87] have shown that the additional loss would help the diffusion network to generate more realistic results. As in [90, 87] a diffusion loss is applied on image patches to improve the quality of the generated image. Following a similar motivation, but with the goal to improve the quality of generated strokes, we propose to train the diffusion model with Dynamic Time Warping loss (DTW).

Dynamic time warping

In general, DTW [9] computes the optimal match between ground-truth (GT) $T = (t_1, t_2, t_3, ..., t_m)$ and predicted sequences $P = (p_1, p_2, p_3, ..., p_n)$ of different lengths by finding the warping path between two sequences. In DTW loss, the cost matrix is calculated as:

$$cost(i, j) = ||p_i - t_j||^2 \tag{6.9}$$

The accumulative cost matrix (A) is given as,

$$A(i, j) = cost(i, j) + min[A(i-1, j), A(i-1, j-1), \tag{6.10}$$

$$A(i, j-1)] \tag{6.11}$$

for $1 \leq i \leq n$ and $1 \leq j \leq m$.

Given matrix A, DTW computes the optimal warping path from $A(n, m)$ to $A(1, 1)$ as an alignment of points in P to points in T expressed as index mapping $\alpha : \{1, \ldots, n\} \rightarrow \{1, \ldots, m\}$, where α is an onto function.

$$\mathcal{L}_{DTW}(P, T) = \sum_{i=1}^{n} ||p_i - t_{\alpha(i)}||. \tag{6.12}$$

We sample the stroke points at each training iteration in the diffusion model with $N{=}60$ denoising steps to predict the stroke sequence. Then we apply DTW loss on each predicted and ground-truth stroke sequence. In our work, we take the sum of both the DTW $\mathcal{L}_{DTW}(P, T)$ and diffusion $\mathcal{L}_{diffusion}$ losses as our final loss $\mathcal{L}_{DTW}(P, T) + \mathcal{L}_{diffusion}$,

In our experience of training the diffusion model with auxiliary DTW loss, it seems that the diffusion model is very vulnerable to diverging if the magnitude of auxiliary loss $(\mathcal{L}_{DTW}(P, T))$ gets more prominent than the diffusion loss $\mathcal{L}_{DTW}$. To prevent model divergence during training, we reduced the learning rate to 10^{-6} so that we can avoid learning divergence.

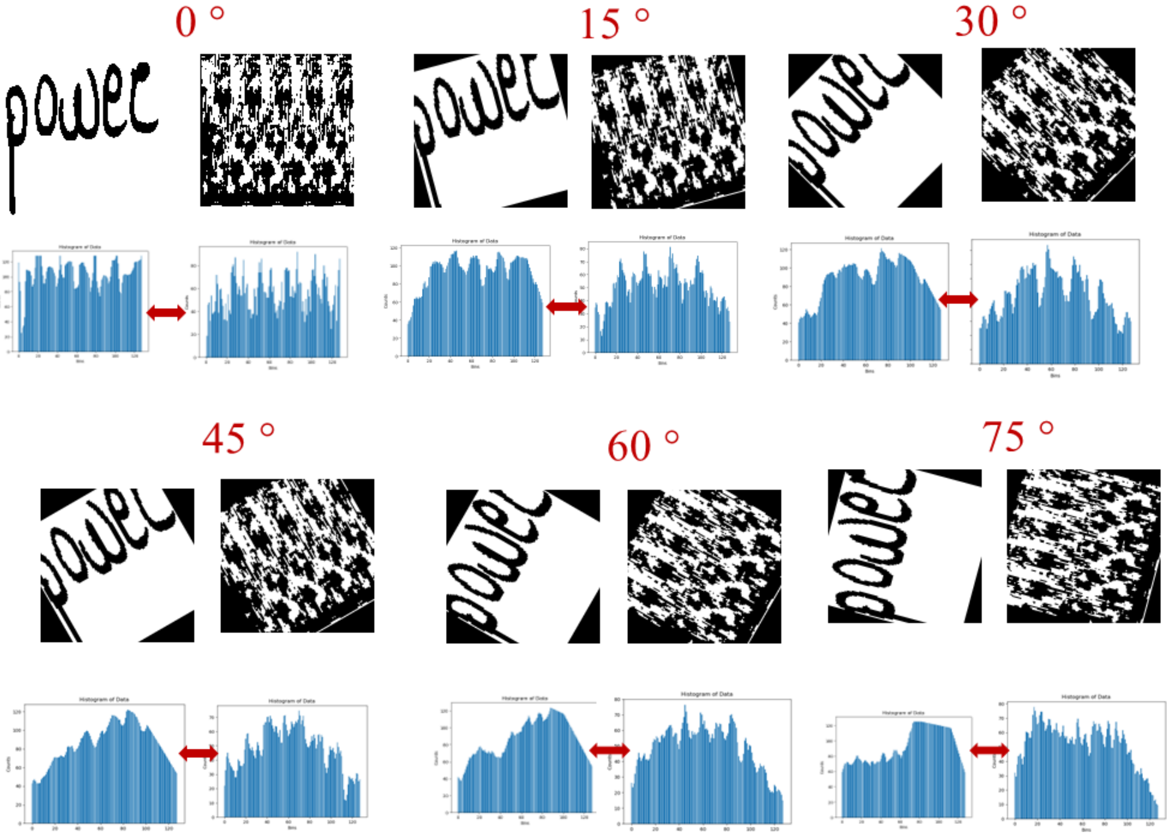


Figure 6.5: A real image and a generated image, and their respective histograms. We show all the six rotated image pairs and their histogram.

Method	$dist_{p,t}$ (mean)	$dist_{p,t}$ (std)	$dist_{t,p}$ (mean)	$dist_{t,p}$ (std)
Stroke Diffusion [53]	0.1513	0.1580	0.1184	0.0646
MS	0.1377	0.1139	0.1116	0.0997
MS_{dtw}	0.1382	0.1121	0.1094	0.1071
MSCAP	**0.1368**	0.1129	0.1050	0.0787
$MSCAP_{dtw}$	0.2184	**0.1050**	**0.0594**	**0.0487**

Table 6.2: The distance of nearest predicted to GT points $dist_{p,t}$ and the distance of nearest GT to predicted points $dist_{t,p}$ for different style features.

Method	IS ↑	FID ↓	PSNR ↑	MSSIM ↑	Style ↓	Projected shape ↓
HiGAN+ [26]	1.594	2.310	11.749	0.6853	0.3390	0.0925
Trace [5]	1.590	0.6515	**17.185**	**0.9030**	0.2923	0.1398
Stroke diffusion [53]	1.572	0.4718	13.220	0.7490	0.3754	0.0658
Our(MS)	**1.595**	0.3895	13.122	0.7563	0.2296	0.0649
Our(MS˙dtw)	1.580	**0.3819**	13.166	0.7583	0.2265	0.0656
Our(MSCAP)	1.570	0.3860	13.177	0.7590	**0.2112**	0.0645
Our(MSCAP˙dtw)	1.560	0.3880	13.192	0.7590	0.2222	**0.0643**

Table 6.3: Long sentences: Quantitative comparison of our method with state-of-the-art methods for handwriting imitation. The style input is an online handwritten image from the IAM-online dataset.

6.2 Experiments

We evaluate our proposed method based on sequence-matching evaluation metrics and generated image quality. We will briefly describe the evaluation metrics used to evaluate the proposed method.

6.2.1 Strokes-based metrics

We used a distance-based evaluation metric to evaluate stroke sequence generation. The average distance of points in ground-truth (T) stroke to its nearest predicted stroke (P) is

Method	IS ↑	FID ↓	PSNR ↑	MSSIM ↑	Style ↓	Projected shape ↓
HiGAN+ [26]	1.862	**0.8052**	**10.9619**	**0.5523**	**0.04425**	0.0833
wordstylist [57]	1.8613	0.9106	10.2720	0.5017	0.0583	0.0869
Strokes diffusion [53]	1.8681	1.939	9.4200	0.5001	0.1061	**0.08135**
Ours(MS)	1.8583	1.960	9.4276	0.5008	0.0967	0.0840
Ours(MS_{dtw})	1.8660	1.959	9.4086	0.5027	0.0982	0.0846
Ours($MSCAP$)	1.8749	1.992	9.4773	0.5111	0.1034	0.08177
Ours($MSCAP_{dtw}$)	**1.8791**	2.0466	9.4778	0.5102	0.1074	**0.08154**

Table 6.4: Words only: Quantitative comparison of our method with state-of-the-art methods for handwriting imitation. The style input is an offline handwritten image from the IAM-offline dataset.

denoted by $dist_{t,p}$. Similarly, the average distance of points in the predicted stroke (P) to its nearest ground-truth stroke (T) is denoted by $dist_{p,t}$. The metric $dist_{t,p}$ signifies that every ground-truth stroke point is close to the predicted point and vice versa for $dist_{p,t}$. $dist_{t,p}$ and $dist_{p,t}$ are the same evaluation metrics as used in [5]. However, apart from the mean $(mean)$ of the distances between predicted and ground truth stroke points, we also compute the standard deviation (std) of the metrics.

6.2.2 Image-based metrics

IS, FID:

In addition to the distance-based metric, we also use perceptive scores such as Inception Score (IS) and Fréchet inception distance (FID) to evaluate the quality of the generated image as used in the previous works [26, 57, 59, 10]. A higher value of IS signifies that the generated images are diverse and contain meaningful and distinct objects. However, IS has limitations and does not compare real and generated images. To compare real and generated images, we use FID, which measures the similarity between the distribution of real and generated images.

FID score is computed with the mean and covariance of IS for real and generated images as follows:

$$\text{FID } = \left\|\mu_{\text{real }}-\mu_{\text{gen }}\right\|^2+\operatorname{Tr}\left(\Sigma_{\text{real }}+\Sigma_{\text{gen }}-2\left(\Sigma_{\text{real}} \Sigma_{\text{gen }}\right)^{1 / 2}\right) \tag{6.13}$$

μ_{real} and μ_{gen} are the means and Σ_{real} and $\Sigma_{gen}$are the covariance matrix of the real and generated image. *Tr* denotes the trace of a covariance matrix. A lower FID score indicates better similarity between the real and generated image distributions. FID is a better measure since it captures both image quality and diversity.

PNSR, MSSIM:

PSNR, or Peak Signal-to-Noise Ratio, is a metric commonly used to measure the quality of reconstructed or generated images. A higher PSNR generally indicates better image quality. The PSNR is calculated using the mean squared error (MSE) between the real and generated images as follows:

$$\mathrm{PSNR}=10 \cdot \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right), \tag{6.14}$$

where MAX is the maximum pixel value of the image (255). A higher PSNR value is desirable as it indicates less perceptual loss in the generated image. However, PSNR relies on mean squared error, which does not always align well with human perception. In some cases, improvements in PSNR may not necessarily correspond to visually more pleasing images.

On the other hand, the Mean Structural Similarity Index (MSSIM) measures the quality of an image in terms of structural information, taking into account luminance, contrast, and structure. These components are combined to provide an overall measure of similarity:

$$\text{MSSIM}(x, y) = \frac{1}{N} \sum_{i=1}^{N} \frac{(2\mu_{x_i}\mu_{y_i} + c_1) \cdot (2\sigma_{x_i y_i} + c_2)}{\left(\mu_{x_i}^2 + \mu_{y_i}^2 + c_1\right) \cdot \left(\sigma_{x_i}^2 + \sigma_{y_i}^2 + c_2\right)} \tag{6.15}$$

Here, μ is the mean, σ is the standard deviation, $\sigma_{x_i y_i}$ is the covariance for the i^{th} and j^{th} local region in image x and y. N is the total number of regions in the images. c_1 and c_2 are small constants to avoid instability when the denominators are close to zero. The MSSIM metric ranges from -1 to 1, where 1 indicates identical images, 0 indicates no similarity, and -1 indicates complete dissimilarity. Since we generate a stroke sequence, we plot the stroke sequence in the form of an image before applying conventional (IS, FID, MSSIM,PSNR) image evaluation methods.

Style distance

The evaluation metrics described above are designed to evaluate the image generation quality for natural scenes. In natural scenes, IS can signify objectness as a fair measure. Similarly, MSSIM accounts for luminance, contrast, and structure comparison, which are useful measures to evaluate the quality of generated images in the case of natural images. However, the conventional metrics [28, 86] such as IS, FID, PSNR, and MSSIM are insufficient to evaluate the quality of handwriting images.

Therefore, we propose to use style features to compute the similarity of the calligraphic style of the real and the generated image. We train the convolutional network based on transformer architecture (Sec. 6.1.1) to learn the handwriting style. The writer ID is used to train the network with cross-entropy loss. The style features are extracted from the fully connected layer before the classification layer. We compute the L1 distance between the style features of the real and generated image. This evaluation metric gives better similarity for the generated images resembling the overall (global) style of the real image. Style distance evaluates closeness in calligraphic style and does not give much attention to background texture. The explicit focus on calligraphy style is missing in conventional metrics (IS, FID, MSSIM, PSNR). The results for online and offline handwriting are listed in Table 6.3 and Table 6.4.

Projection character shape matching

The calligraphic style of the handwriting is captured by the evaluation measure described in Section 6.2.2. In this section, we introduce a more granular measure to evaluate the

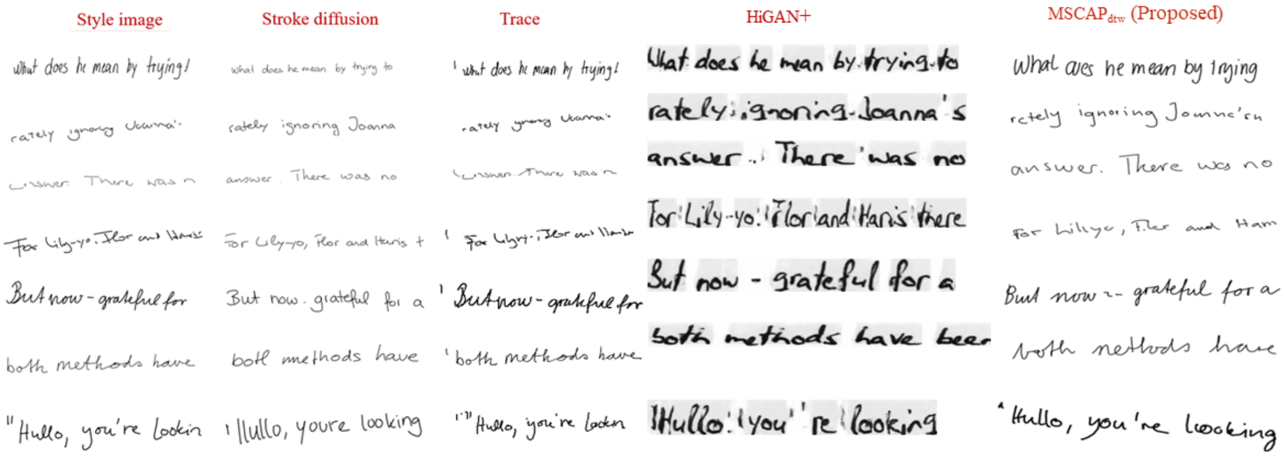


Figure 6.6: Visual comparison with previous image generation methods.

local character shape in addition to the overall calligraphic style of the handwriting. In our work, we propose to use the distribution of pixel histograms in real and generated images to calculate the similarity between local character shapes.

The previous handwriting imitation work [26] suggests the L1-distance between ground-truth x and generated $\hat{x}$ image to calculate the similarity between two images. The L1 distance works well for global image quality since it takes the average of the difference of pixels in two images. However, it does not seem sensitive to the individual characters' shape in handwriting, so we introduce a projection character shape matching.

To calculate the character shape matching, we measure the distribution distances on multiple 1D projections as shown in Fig 6.5. We first binarize the images, then we construct the histogram of pixel distribution along the x-axis for real and generated images at six different orientations (0, 15, 30, 45, 60, and 75). In Fig. 6.5, we show the generated images before training the diffusion model to highlight the clear difference between the histogram distribution of real and generated images.

We use KL-divergence to measure the distance of the distribution of the foreground pixels in the generated $\hat{x}$ and ground-truth x images at the p^{th} orientation as follows.

$$\mathcal{L}_{pc-kl}(\boldsymbol{Y},\hat{\boldsymbol{Y}}) = \frac{1}{P}\sum_{p=1}^{P}\mathbf{KL}\left(\frac{\phi_p(\boldsymbol{Y})}{\sum\phi_p(\boldsymbol{Y})},\frac{\phi_p(\hat{\boldsymbol{Y}})}{\sum\phi_p(\hat{\boldsymbol{Y}})}\right) \tag{6.16}$$

where $\phi_p(\boldsymbol{Y})$ and $\phi_p(\hat{\boldsymbol{Y}})$ is the distribution of foreground pixels in p^{th} orientation. To the best of our knowledge, our work is the first to introduce projection character shape matching to evaluate the quality of the calligraphic shapes of the character in handwriting image generation.

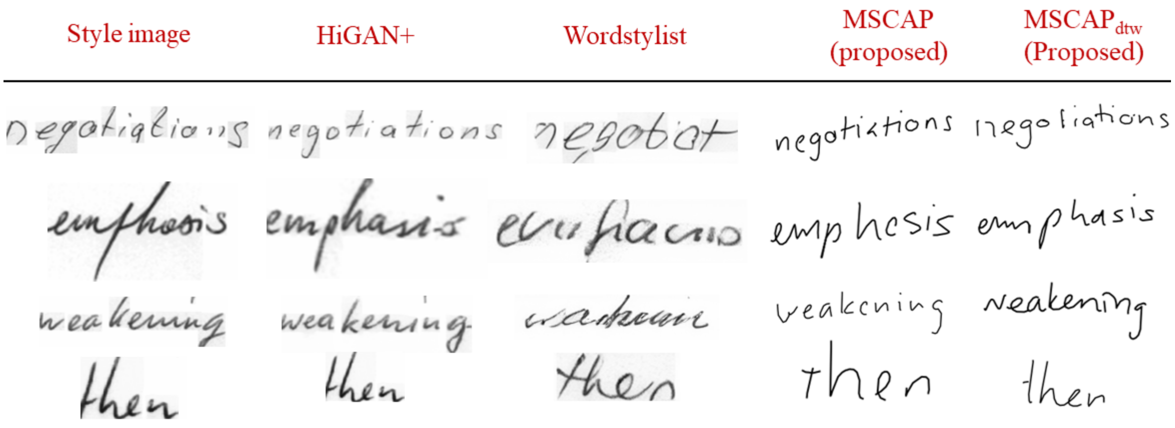


Figure 6.7: Visual comparison of the proposed method with the state-of-the-art handwriting image generation methods for offline handwriting word samples.

6.2.3 Results

Handwriting stroke prediction

We utilized IAM-online [54] dataset to train our diffusion network. It includes the images of handwritten text, the textual content in the image as a string of characters, and the x, and y coordinates of the strokes with a pen-up and down information. The previous method for handwriting image generation via stroke generation [53] compares the results only for the quality of the generated image (Sec. 6.2.2). We present a comprehensive analysis to evaluate the quality of the handwriting stroke generation network with the set of metrics described in Sec 6.2.1.

Table 6.2 compares the distance between predicted and ground-truth stroke sequences. We can see that the proposed multi-scale attention network with character pair features outperforms image features extracted from network [39] trained on Imagenet [46]. In our experimentation, *MS* gives better results than the baseline [53] for both $dist_{p,t}$ (mean) and $dist_{t,p}$ (mean). Applying DTW loss and diffusion loss with $MSCAP$ improves the results compared to *MS*. In addition, applying DTW loss with $MSCAP_{dtw}$ gives excellent results for $dist_{t,p}$ (mean). On the other hand, $MSCAP_{dtw}$ gives a slightly higher value for $dist_{p,t}$ (mean), as DTW loss may introduce stray points in predicted stroke sequence as highlighted in previous work [34]. This is the drawback of DTW loss since it incurs the loss by finding the accumulative cost matrix as shown in Equation 6.10. Apart from the occasional introduction of stray points, the overall quality of stroke generation improves with DTW loss as shown in Fig. 6.6 and discussed in detail later in Sec 6.2.4.

Handwriting image generation

To evaluate the quality of image generation, we divide our analysis into two scenarios, online and offline input sample image. In the first scenario of online handwriting images, we provide the style image generated via stroke generation. These images have no texture and only contain black handwriting on a white background. Sample input style images are shown in the leftmost column in Fig. 6.6. In the second scenario, the offline handwriting image serves as a style image. Offline handwriting images may contain handwriting background and may contain words with variable font thickness as shown in the leftmost column of Fig. 6.7.

We evaluate our proposed method for online handwriting as listed in Table 6.3. We evaluate our method based on conventional image generation metrics and proposed evaluation metrics (style distance and projected character matching). HiGAN+ [26], which generates state-of-the-art results for handwriting image generation, seems to have failed to imitate online sample images. It does not give satisfactory results based on conventional methods (IS, FID, PSNR, MSSIM) as well as the proposed evaluation metrics. The poor performance of HiGAN+ on online images might be because it over-fitted during training to predict texture as well, even though there is no texture in the online style images.

On the other hand, both Trace [5] and stroke diffusion [53] are stroke sequence prediction networks. There is no issue with background texture in generated handwriting for them. They both produce satisfactory results with Trace even better in terms of *PSNR* and *MSSIM.* The style distance is reasonable for Trace [5] but it produces a larger projected character matching error. The lack of character shape matching is because the Trace model is based on LSTM [27] network and trained only with DTW loss. The DTW loss is helpful for overall style matching but does not give sufficient importance to local character shapes. Stroke diffusion model [53] produces good results for conventional and proposed metrics, except it cannot replicate style well as shown in Table 6.3(style). The stroke diffusion does not follow the style template because its style features are trained on natural images [39]. The proposed multi-scale style features with DTW loss MS_{dtw} give the lowest FID scores with low values of style distance and projection character shape matching error.

Our proposed method for multi-scale character pair features with DTW loss $MSCAP_{dtw}$, improves the results of FID scores, style distance, and Projected character shape matching. The multi-scale attention-based features for character pairs $MSCAP$ produce reasonable style. The DTW loss further improves the projected character shape matching error which strengthens our claim that DTW loss along with diffusion loss could improve overall handwriting stroke generation.

The qualitative examples in Fig. 6.6 show that the stroke generation with proposed methods produces better stroke generation as compared to LSTM architecture [5] (Trace), stroke diffusion model [53], and HiGAN+ [26]. [53] extract style features from [39] trained

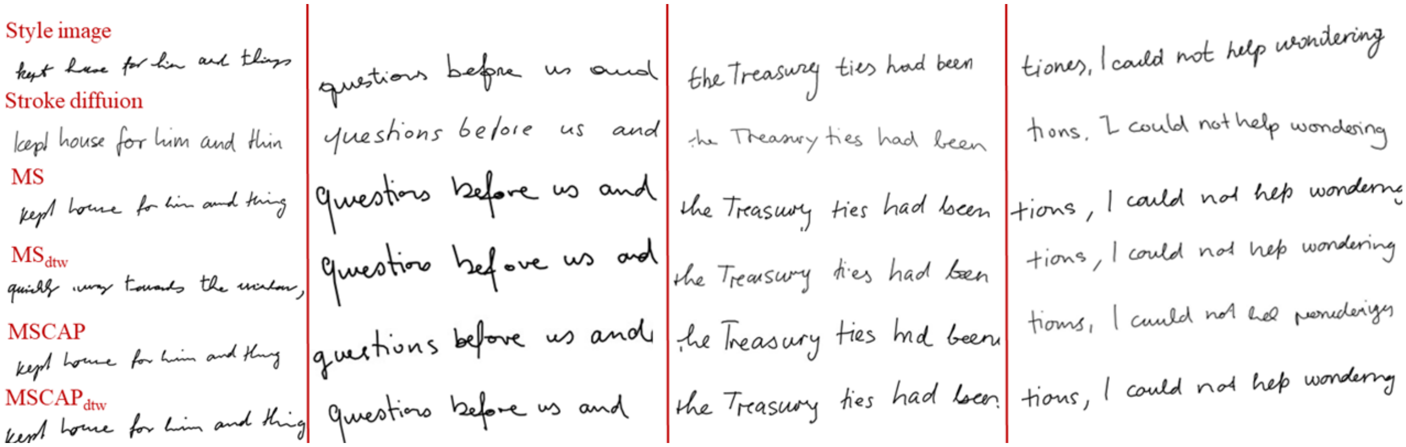


Figure 6.8: Visual comparison of multi-scale attention style features and the Dynamic Time Warping (DTW) loss.

Method	# of Params	Training time	Writer ID	Refinment	Recognition	Unseen style	Strokes
Proposed(MSCAP)	16.8 M	6 hours				✓	✓
Proposed($MSCAP_{dtw}$)	16.8 M	24 hours				✓	✓
Wordstylist	40.4 M	7 days	✓				
HiGAN+	14.0 M	3 days	✓	✓	✓		

Table 6.5: Comparison of model size and architecture with the state-of-the-art methods.

on natural images. The Trace [5] also extracts style features from resnet layer before feeding it into LSTM architecture for stroke prediction. Whereas [57] does not extract style features from the images. Rather, it learns the style from integer input for writer ID. Therefore, [57] shows the least generalization and style similarity and our method $MSCAP_{dtw}$ shows the highest style similarity.

To depict the generalization ability of our methods as compared to previous methods, we compute the evaluation metrics on offline handwriting images. These images differ from online images in terms of background texture, writing styles, and font thickness. The offline and online IAM datasets [54] are composed of words and lines of text, respectively, which is another prominent difference between them. Since previous methods [26, 57] are trained on images of words from IAM-online datasets, we also evaluated our method against them using the same offline word images. These word images from the IAM-offline dataset are completely unseen for our proposed diffusion model. It can be seen from Table 6.4 that the HiGAN produces good results for all the evaluation metrics except for the projected character shape matching. The reason for the extraordinary results of HiGAN+ [26] on the offline datasets and its inability to generate any reasonable images in the online dataset (Table 6.3) suggests that the network lacks generalization ability. It may work only on offline words, which restricts its capability to diverse style images. [53] produces reasonable results; however, it cannot imitate style since its features are trained on natural images. Our proposed method is trained only on online images [54], but it can still produce competitive results for offline image samples. It shows that our diffusion model has better generalization ability than GAN architecture [26].

6.2.4 Discussion

Our multi-scale style feature extraction can imitate style from template images. The addition of DTW loss with *MSCAP* gives us the best (lowest) projected character shape matching error as shown in the last column of Table 6.4 (projected shape). For the qualitative examples shown in Fig. 6.7 the state-of-the-art HiGAN+ [26] produces nearly perfect results in the case of offline sample images, but HiGAN+ has poor generalizability since it does not perform reasonably on online sample images. On the other hand, [57] does not extract style features from the images. Rather, it learns the style from integer input for writer ID. Therefore, [57] shows the least generalization and style similarity. Our method shows the highest style similarity as well as generalizability. Notably, [57] cannot process multiple words without modifying input interfacing. However, the proposed diffusion-based handwriting image generation method can generate short and long text without additional effort.

We also compared visual results for variants *MS*, MS_{dtw}, *MSCAP*, and $MSCAP_{dtw}$ of our method in Fig. 6.8. We observe that the character shapes are good for all variants, but the overall style is better imitated for *MSCAP* as we focus on character pair features. We can also validate in Fig 6.8 that the stroke generated with the diffusion model produces readable text even though we have not leveraged any text recognition module.

Some qualitative results of our method $MSCAP_{dtw}$ in comparison with the previous methods [26, 53] are shown in Fig. 6.9. Here, we stress the capability of our method to perform well on unseen text content. We can see that our method ($MSCAP_{dtw}$) can imitate the unseen text well as compared to stroke diffusion [53]. Moreover, our method also performs well compared to the state-of-the-art image generation method (HiGAN+ [26]).

Some of the previous methods [57, 10, 59] can only perform inference on single words as textual input; therefore they cannot be applied to long sentences. However, our stroke generation method does not have such restriction and it applies to long sentences as a textual input.

Finally, we highlight the training time and auxiliary networks used in previous methods [26, 53] and our proposed method in Table 6.5. HiGAN+ [26] utilizes text recognition, patch refinement, and writer ID module. It takes 3 days on a single NVIDIA A100 GPU. The diffusion model for handwriting image generation [53], does not include a text recognition module but it still takes a longer time to train due to the iterative learning of diffusion networks. Our proposed method *MSCAP* only takes 6 hours to complete 60k iterations to converge the learning of stroke generation with the diffusion model. Our model takes significantly less time since we generate strokes rather than images; the number of predicted strokes is much smaller than the number of pixels in the image. In addition, adding DTW loss $MSCAP_{dtw}$ increases the training time to around 24 hours because we perform sampling during training.

6.3 Conclusion

We have demonstrated that the diffusion model conditioned on multi-scale character pair features improves the calligraphic style imitation for handwriting stroke generation. Importantly, the proposed method $MSCAP_{dtw}$ not only outperforms stroke generation from online sample images, but it also produces competitive results for unseen offline sample images. We also introduce evaluation metrics for handwriting image quality evaluation based on calligraphic style and character shape matching. Our quantitative and qualitative analysis also suggest that effective style features $MSCAP$ features help the diffusion model with efficient features to imitate handwriting style. This work with the diffusion model would provide the foundations for future work to generate handwriting strokes for arbitrary text in any calligraphic style.

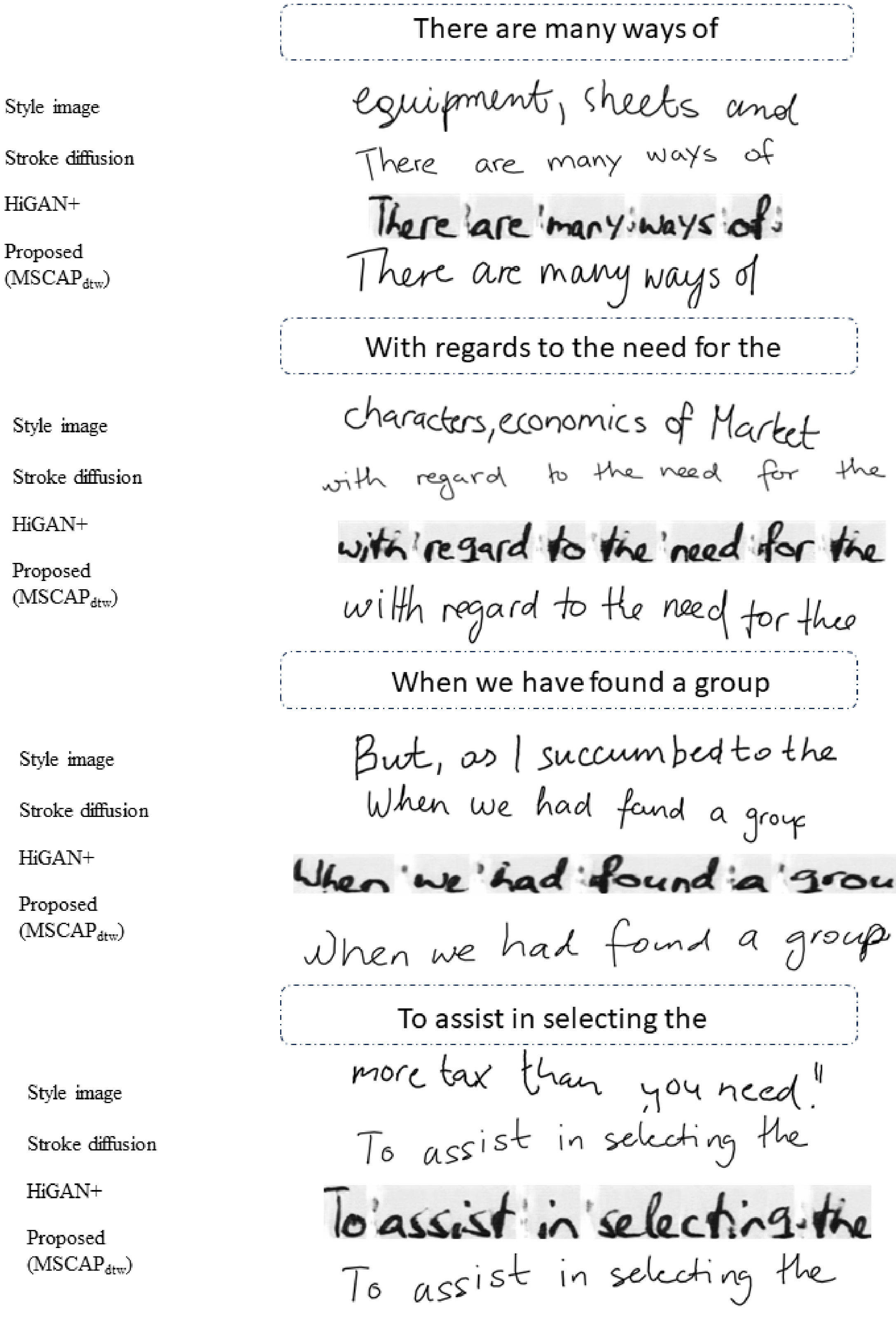


Figure 6.9: Exemplar qualitative results of image generation through stroke generation on the IAM-online dataset. For each example, the text content is shown in a dotted rectangular block with the input style image and generated images below it.

Bibliography

[1] Akmal Butt, M., Maragos, P.: Optimum design of chamfer distance transforms. IEEE Transactions on Image Processing **7**(10), 1477–1484 (1998). https://doi.org/10.1109/83.718487

[2] Aksan, E., Pece, F., Hilliges, O.: Deepwriting: Making digital ink editable via deep generative modeling. In: Proceedings of the 2018 CHI conference on human factors in computing systems. pp. 1–14 (2018)

[3] Alberti, M., Vögtlin, L., Pondenkandath, V., Seuret, M., Ingold, R., Liwicki, M.: Labeling, cutting, grouping: an efficient text line segmentation method for medieval manuscripts. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1200–1206. IEEE (2019)

[4] Alonso, E., Moysset, B., Messina, R.: Adversarial generation of handwritten text images conditioned on sequences. In: 2019 international conference on document analysis and recognition (ICDAR). pp. 481–486. IEEE (2019)

[5] Archibald, T., Poggemann, M., Chan, A., Martinez, T.: Trace: A differentiable approach to line-level stroke recovery for offline handwritten text. arXiv preprint arXiv:2105.11559 (2021)

[6] Axler, G., Wolf, L.: Toward a dataset-agnostic word segmentation method. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 2635–2639. IEEE (2018)

[7] Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9365–9374 (2019)

[8] Barakat, B.K., El-Sana, J., Rabaev, I.: The pinkas dataset. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 732–737. IEEE (2019)

[9] Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD workshop. vol. 10, pp. 359–370. Seattle, WA, USA: (1994)

[10] Bhunia, A.K., Khan, S., Cholakkal, H., Anwer, R.M., Khan, F.S., Shah, M.: Handwriting transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1086–1094 (2021)

[11] Bhunia, A.K., Bhowmick, A., Bhunia, A.K., Konwer, A., Banerjee, P., Roy, P.P., Pal, U.: Handwriting trajectory recovery using end-to-end deep encoder-decoder network. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3639–3644. IEEE (2018)

[12] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)

[13] Boillet, M., Kermorvant, C., Paquet, T.: Multiple document datasets pre-training improves text line detection with deep neural networks. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 2134–2141. IEEE (2021)

[14] Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)

[15] Cheng, S.I., Chen, Y.J., Chiu, W.C., Tseng, H.Y., Lee, H.Y.: Adaptively-realistic image generation from stroke and sketch with diffusion model. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4054–4062 (2023)

[16] Davis, B., Tensmeyer, C., Price, B., Wigington, C., Morse, B., Jain, R.: Text and style conditioned gan for generation of offline handwriting lines. arXiv preprint arXiv:2009.00678 (2020)

[17] Davis, B., Tensmeyer, C., Price, B., Wigington, C., Morse, B., Jain, R.: Text and style conditioned gan for generation of offline handwriting lines. arXiv preprint arXiv:2009.00678 (2020)

[18] Demır, A.A., ÖzŞeker, İ., Özkaya, U.: Text line segmentation in handwritten documents with generative adversarial networks. In: 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). pp. 1–5. IEEE (2021)

[19] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)

[20] Diaz, M., Crispo, G., Parziale, A., Marcelli, A., Ferrer, M.A.: Writing order recovery in complex and long static handwriting (2022)

[21] Faundez-Zanuy, M., Fierrez, J., Ferrer, M.A., Diaz, M., Tolosana, R., Plamondon, R.: Handwriting biometrics: Applications and future trends in e-security and e-health. Cognitive Computation **12**(5), 940–953 (2020)

[22] Fogel, S., Averbuch-Elor, H., Cohen, S., Mazor, S., Litman, R.: Scrabblegan: Semi-supervised varying length handwritten text generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4324–4333 (2020)

[23] Gader, T.B.A., Echi, A.K.: Unconstrained handwritten arabic text-lines segmentation based on ar2u-net. In: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 349–354. IEEE (2020)

[24] Gader, T.B.A., Echi, A.K.: Deep learning-based segmentation of connected components in arabic handwritten documents. In: International Conference on Intelligent Systems and Pattern Recognition. pp. 93–106. Springer (2022)

[25] Gan, J., Wang, W.: Higan: Handwriting imitation conditioned on arbitrary-length texts and disentangled styles. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 7484–7492 (2021)

[26] Gan, J., Wang, W., Leng, J., Gao, X.: Higan+: Handwriting imitation gan with disentangled representations. ACM Transactions on Graphics (TOG) **42**(1), 1–17 (2022)

[27] Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: Lstm: A search space odyssey. IEEE transactions on neural networks and learning systems **28**(10), 2222–2232 (2016)

[28] Gu, S., Bao, J., Chen, D., Wen, F.: Giqa: Generated image quality assessment. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 369–385. Springer (2020)

[29] Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)

[30] Guyon, I., Schomaker, L., Plamondon, R., Liberman, M., Janet, S.: Unipen project of on-line data exchange and recognizer benchmarks. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5). vol. 2, pp. 29–33. IEEE (1994)

[31] Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al.: A survey on vision transformer. IEEE transactions on pattern analysis and machine intelligence **45**(1), 87–110 (2022)

[32] Han, Z., Wang, Y., Zhou, L., Wang, P., Yan, B., Zhou, J., Wang, Y., Shen, D.: Contrastive diffusion model with auxiliary guidance for coarse-to-fine pet reconstruction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 239–249. Springer (2023)

[33] Hanif, S., Latecki, L.J.: Autonomous character region score fusion for word detection in camera-captured handwriting documents

[34] Hanif, S., Latecki, L.J.: Strokes trajectory recovery for unconstrained handwritten documents with automatic evaluation (2023)

[35] Hanif, S., Li, C., Alazzawe, A., Latecki, L.J.: Image retrieval with similar object detection and local similarity to detected objects. In: Pacific Rim International Conference on Artificial Intelligence. pp. 42–55. Springer (2019)

[36] Hanif, S., Li, C., Alazzawe, A., Latecki, L.J.: Image retrieval with similar object detection and local similarity to detected objects. In: Pacific Rim International Conference on Artificial Intelligence. pp. 42–55. Springer (2019)

[37] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[38] Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. The Journal of Machine Learning Research **23**(1), 2249–2281 (2022)

[39] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

[40] Hu, H., Zhang, C., Luo, Y., Wang, Y., Han, J., Ding, E.: Wordsup: Exploiting word annotations for character based text detection. In: Proceedings of the IEEE international conference on computer vision. pp. 4940–4949 (2017)

[41] Jemni, S.K., Kessentini, Y., Kanoun, S.: Out of vocabulary word detection and recovery in arabic handwritten text recognition. Pattern Recognition **93**, 507–520 (2019)

[42] Kang, L., Riba, P., Rusinol, M., Fornes, A., Villegas, M.: Content and style aware generation of text-line images for handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(12), 8846–8860 (2021)

[43] Kang, L., Riba, P., Wang, Y., Rusinol, M., Fornés, A., Villegas, M.: Ganwriting: content-conditioned generation of styled handwritten word images. In: Computer

Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16. pp. 273–289. Springer (2020)

[44] Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5148–5157 (2021)

[45] Kotani, A., Tellex, S., Tompkin, J.: Generating handwriting via decoupled style descriptors. In: European Conference on Computer Vision. pp. 764–780. Springer (2020)

[46] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

[47] Kumar, R., Singh, A.: Detection and segmentation of lines and words in gurmukhi handwritten text. In: 2010 IEEE 2nd International Advance Computing Conference (IACC). pp. 353–356. IEEE (2010)

[48] Kurar Barakat, B., Cohen, R., Droby, A., Rabaev, I., El-Sana, J.: Learning-free text line segmentation for historical handwritten documents. Applied Sciences **10**(22), 8276 (2020)

[49] Lee, A.W., Chung, J., Lee, M.: Gnhk: A dataset for english handwriting in the wild. In: International Conference on Document Analysis and Recognition. pp. 399–412. Springer (2021)

[50] Li, M., Lv, T., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: Trocr: Transformer-based optical character recognition with pre-trained models. arXiv preprint arXiv:2109.10282 (2021)

[51] Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: Casia online and offline chinese handwriting databases. In: 2011 International Conference on Document Analysis and Recognition. pp. 37–41. IEEE (2011)

[52] Lu, M., Zhou, W., Ji, R.: Automatic scoring system for handwritten examination papers based on yolo algorithm. In: Journal of Physics: Conference Series. vol. 2026, p. 012030. IOP Publishing (2021)

[53] Luhman, T., Luhman, E.: Diffusion models for handwriting generation. arXiv preprint arXiv:2011.06704 (2020)

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

[55] Nguyen, H.T., Nakamura, T., Nguyen, C.T., Nakawaga, M.: Online trajectory recovery from offline handwritten japanese kanji characters of multiple strokes. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 8320–8327. IEEE (2021)

[56] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)

[57] Nikolaidou, K., Retsinas, G., Christlein, V., Seuret, M., Sfikas, G., Smith, E.B., Mokayed, H., Liwicki, M.: Wordstylist: Styled verbatim handwritten text generation with latent diffusion models. arXiv preprint arXiv:2303.16576 (2023)

[58] Nishide, S., Okuno, H.G., Ogata, T., Tani, J.: Handwriting prediction based character recognition using recurrent neural network. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics. pp. 2549–2554. IEEE (2011)

[59] Pippi, V., Cascianelli, S., Cucchiara, R.: Handwritten Text Generation from Visual Archetypes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)

[60] Plamondon, R., Srihari, S.N.: Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on pattern analysis and machine intelligence **22**(1), 63–84 (2000)

[61] Privitera, C.M., Plamondon, R.: A system for scanning and segmenting cursively handwritten words into basic strokes. In: Proceedings of 3rd International Conference on Document Analysis and Recognition. vol. 2, pp. 1047–1050. IEEE (1995)

[62] Rabhi, B., Elbaati, A., Boubaker, H., Hamdi, Y., Hussain, A., Alimi, A.M.: Multilingual character handwriting framework based on an integrated deep learning based sequence-to-sequence attention model. Memetic Computing **13**(4), 459–475 (2021)

[63] Rabhi, B., Elbaati, A., Boubaker, H., Pal, U., Alimi, A.: Multi-lingual handwriting recovery framework based on convolutional denoising autoencoder with attention model (2022)

[64] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)

[65] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)

[66] Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)

[67] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)

[68] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

[69] Ryu, J., Koo, H.I., Cho, N.I.: Word segmentation method for handwritten documents based on structured learning. IEEE Signal Processing Letters **22**(8), 1161–1165 (2015)

[70] Santoso, R., Suprapto, Y.K., Yuniarno, E.M.: Kawi character recognition on copper inscription using yolo object detection. In: 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM). pp. 343–348. IEEE (2020)

[71] Senatore, R., Santoro, A., Parziale, A., Marcelli, A.: A biologically inspired approach for recovering the trajectory of off-line handwriting (2022)

[72] Signed: (2022), `www.signedcards.com`

[73] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[74] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)

[75] Surinta, O., Holtkamp, M., Karaaba, M.F., van Oosten, J., Schomaker, L.R.B., Wiering, M.A.: A* path planning for line segmentation of handwritten documents. In: Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on. pp. 175–180. IEEE (Sep 2014). https://doi.org/http://dx.doi.org/10.1109/ICFHR.2014.37

[76] Surinta, O., Holtkamp, M., Karabaa, F., Van Oosten, J.P., Schomaker, L., Wiering, M.: A path planning for line segmentation of handwritten documents. In: 2014 14th international conference on frontiers in handwriting recognition. pp. 175–180. IEEE (2014)

[77] Uehara, K., Sakanashi, H., Nosato, H., Murakawa, M., Miyamoto, H., Nakamura, R.: Object detection of satellite images using multi-channel higher-order local autocorrelation. In: 2017 IEEE international conference on systems, man, and cybernetics (SMC). pp. 1339–1344. IEEE (2017)

[78] Varga, T., Bunke, H.: Tree structure for word extraction from handwritten text lines. In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05). pp. 352–356. IEEE (2005)

[79] Viard-Gaudin, C., Lallican, P.M., Knerr, S.: Recognition-directed recovering of temporal information from handwriting images. Pattern Recognition Letters **26**(16), 2537–2548 (2005)

[80] Viard-Gaudin, C., Lallican, P.M., Knerr, S., Binter, P.: The ireste on/off (ironoff) dual handwriting database. In: Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318). pp. 455–458. IEEE (1999)

[81] Wang, L., Li, R., Shi, H., Sun, J., Zhao, L., Seah, H.S., Quah, C.K., Tandianus, B.: Multi-channel convolutional neural network based 3d object detection for indoor robot environmental perception. Sensors **19**(4), 893 (2019)

[82] Wei, X.S., Xie, C.W., Wu, J.: Mask-cnn: Localizing parts and selecting descriptors for fine-grained image recognition. arXiv preprint arXiv:1605.06878 (2016)

[83] Wilkinson, T., Brun, A.: A novel word segmentation method based on object detection and deep learning. In: International Symposium on Visual Computing. pp. 231–240. Springer (2015)

[84] Wilkinson, T., Lindstrom, J., Brun, A.: Neural ctrl-f: segmentation-free query-by-string word spotting in handwritten manuscript collections. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4433–4442 (2017)

[85] Wu, Y., Hu, Y., Miao, S.: Object detection based handwriting localization. In: International Conference on Document Analysis and Recognition. pp. 225–239. Springer (2021)

[86] Yang, J., Lyu, M., Qi, Z., Shi, Y.: Deep learning based image quality assessment: A survey. Procedia Computer Science **221**, 1000–1005 (2023)

[87] Yang, S., Hwang, H., Ye, J.C.: Zero-shot contrastive loss for text-guided diffusion image style transfer. arXiv preprint arXiv:2303.08622 (2023)

[88] Yavariabdi, A., Kusetogullari, H., Celik, T., Thummanapally, S., Rijwan, S., Hall, J.: Cardis: A swedish historical handwritten character and word dataset. IEEE Access **10**, 55338–55349 (2022). https://doi.org/10.1109/ACCESS.2022.3175197

[89] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 558–567 (2021)

[90] Zhu, Y., Wu, Y., Olszewski, K., Ren, J., Tulyakov, S., Yan, Y.: Discrete contrastive diffusion for cross-modal and conditional generation. arXiv preprint arXiv:2206.07771 (2022)