Generative Models for Handwriting Synthesis And Imitation

Research Proposal

School of Computer Science & Applied Mathematics
University of the Witwatersrand

Rifumo Mzimba
1619542

Supervised by Dr R. Klein

July 17, 2022

A proposal submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Bachelor of Science with Honours

Abstract

Handwriting synthesis is the process of training a machine to produce text that looks handwritten by a human. Different from script fonts, the characters need to show randomness that exists in real handwriting. Handwriting has been used for many years as a biometric system due to its unique features. A human cannot write the same text in the same way. However, there are unique traits that an individual's handwriting always has. These are influenced by genes (bone structure, hand-eye coordination, gender), practice, the environmental setup (writing equipment, writing surface), and other factors like age, mood, and attitude [Hilton 1992]. Graphology is the study of handwriting. They can identify the writer's emotional state and the personality from just their handwritten text [Beyerstein and Beyerstein 1992].

While graphology extracts the unique features of an individual's handwriting to identify them, handwriting imitation uses these features to mimic a person's handwriting. Writer imitation is a developing field taking handwriting generation a step further. Most research uses deep generative techniques, most noticeably, GANs. There is a lot of room for improvement, and this proposal aims to fill some of the space. We propose using a VAE-GAN hybrid in combination with a state-of-the-art proposal that adds spatial-temporal information to offline handwritten text.

Declaration

I, Rifumo Mzimba, hereby declare the contents of this research proposal to be my own work. This proposal is submitted for the degree of Bachelor of Science with Honours in Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

Contents

List of Figures

Chapter 1

Introduction

Text written by hand has existed for centuries as a means of communication [Aksan *et al.* 2018]. Penmanship or handwriting is a skill learned at an early age by humans [Ghosh *et al.* 2017]. Different styles of writing are developed through practice [Kumar *et al.* 2018a]. Handwriting is considered a form of art that has beneficial impacts on note-taking, writing, and in short- and long-term memory [Aksan *et al.* 2018]. It can further be used for identification due to it being distinct from one person to the next [Kumar *et al.* 2018a].

Due to a digital era shift, handwriting is a fading practice, with schools in North America opting out in teaching it and Finland deeming it not a necessity for daily life [Karavanidou 2017]. In an experiment conducted by Brown [1988], they found that experienced typists can type at speeds of over five words per minute (wpm) faster compared to writing by hand. Since it is feasible to mimic a person's handwriting [Ghosh *et al.* 2017] we can train a generative model to synthesize a person's handwriting style from a given sample.

The model will have the efficiency of typing and relevance in the digital era while maintaining the peculiarness of handwriting styles. Haines *et al.* [2016] show the practical uses of such a model, which includes sending personalized gift messages, giving the avatar in virtual reality games the player's handwriting, using a celebrity's handwriting in movies, and writing personalized books using the handwriting of historical authors. The need to print, fill and scan a form can be completely eliminated. Invitation letters can be personalized, the same goes for emails, even social media posts. The response rate of handwritten text has been shown to be more than double that of typed text [Haines *et al.* 2016]. The breakthrough in this can be transferred to other pattern generation problems.

Handwriting generation is an issue which various researchers have addressed using complex mathematical methods, but the results look synthetic [Kumar *et al.* 2018a]. A key influence came from Graves [2013], who proposed a recurrent neural network (RNN) based generator, where they treat the problem as a sequence generation. The model managed to generate natural-looking text but the styles of the model are limited to samples in the training set. It cannot mimic a specific handwriting. The increase in computational power and the development of generative adversarial networks (GANs) [Goodfellow *et al.* 2014] has attracted the attention of more research into generative models [Turhan and Bilge 2018]. Gonog and Zhou [2019] report that GANs have been successfully used in generative tasks such as cartoon imaging, shoe design, handwriting profiling, generation of images using text description, sensing dark energy in galaxies using gravitational lenses, missing data imputation, enhancement and denoising of speech, to list but a few.

Online handwriting generation has been more successful due to the availability of temporal information. In this research, we propose to tackle the challenges that are faced with offline handwriting generation. Our proposal is highly influenced by novel ideas from Mayr *et al.* [2020]'s work. They add spatial-temporal information to the offline data to make it behave like it was online data. After, they use an online generator for synthesis. Their biggest challenge is that their online approximation sometimes produces incomplete skeletons, reducing the authenticity of the images generated.

There has also been good success with word generation [Fogel *et al.* 2020; Kang *et al.* 2020]. However, their approach does not guarantee that the models will learn the influence handwritten characters have on each other. We thus propose a model that combines the advancements of the two approaches. Our contributions are as follows: imitate a given offline handwriting sample using a VAE-GAN model combined with the spatial-temporal information generated from Mayr *et al.* [2020]'s work.

We also propose robust evaluation techniques that better evaluate the generated results. We restrict our experiments to Latin scripts, specifically English. We will also generate word images and not full paragraphs or sentences.

The rest of the paper is sectioned as follows. Chapter 2 provides a literature survey that this proposal builds upon and background information that makes the proposed work easier to understand. It starts off by a brief review of handwriting recognition techniques that have been proposed. This is covered in Section 2.2. It is divided into handwriting recognition (Section 2.2.1) and writer recognition (Section 2.2.2). Afterward, we provide a survey on handwriting generation and imitation in Section 2.3. The section starts off by providing a background on generative models in Section 2.3.1. Following it is a related work sub-section on state-of-the-art handwritten text synthesis in Section 2.3.2. Generative models used in other fields are concisely discussed in Section 2.3.3. The approaches used serve as inspiration for this domain.

After proving background knowledge and a full survey on the work done in the handwriting generation domain, Chapter 3 details the methodology that we will follow for this research. We provide our hypothesis in Section 3.2 which will be accepted or rejected at the end of this proposed research. Research questions are provided in Section 3.3. These guide the experiments we will be conducting (Section 3.4.3) and the evaluation techniques we will employ (Section 3.4.3). The datasets we will be using are found in Section 3.4.1. The full details of the proposed model can be found at Section 3.4.2.

Chapter 4 examines the feasibility of the research. We provide the time-line this research will follow in Section 4.2 and an analysis of the possible risks that might affect our research in Section 4.3. Finally, Chapter 5 presents the summary and conclusion of our research proposal.

Chapter 2

Literature Review

2.1 Introduction

Chapter 1 introduces us to the handwriting domain by providing background and spur for handwriting synthesis and imitation. We also saw that there is more research and success in generative models spanning over multiple fields. In this section, we dive deeper into generative models, specifically used in the handwriting domain. Section 2.3 briefly reviews handwriting recognition as the inverse problem to handwriting generation. This is also a sub-problem for an automated generative system that does not use annotated data. Section 2.2.2 discusses writer identification which can be used to evaluate the success of handwriting imitation.

A survey on recognition follows in Section 2.3. The section starts off with Section 2.3.1 which provides background on architectures that are used for generative pursuits. The methods are divided into Boltzmann Machines, Generative Adversarial Networks, Autoencoder, and Autoregressive networks. Section 2.3.2 reviews work done to synthesize realistic handwriting. It is divided into Online and Offline synthesis. The subsection provides the achievements and challenges of the cutting-edge proposals. Generative and imitative tasks in other fields are discussed in Section 2.3.3. Publicly available handwriting datasets are explored in Section 2.4. Finally, Section 2.5 summarises and concludes the literature survey.

2.2 Recognition

The problem of handwriting imitation can be broken down into two major components: handwriting recognition and handwriting generation. Handwriting recognition is the encoding of input text from the digitizer, pen position sensor, images or scanned documents into computer interpretable format [Memon *et al.* 2020]. It has been explored for over forty years [Elanwar 2013]. It is classified into offline and online handwriting recognition depending on how it was acquired. If the handwriting was scanned, it is referred to as offline whereas if it was recognized using a stylus pen on a touchpad it is referred to as online recognition [Purohit and Chauhan 2016].

Cursive handwriting has been a challenge since it cannot be segmented into letters. Offline recognition is harder compared to online recognition due to varying line thickness and background textures [Alonso *et al.* 2019]. State-of-the-art optical character recognition (OCR) can handle offline data but struggle with ill-written texts and background noise [Fogel *et al.* 2020].

2.2.1 Handwriting Recognition

The inverse problem of handwriting generation is handwriting recognition. Optical character recognition (OCR) is the process that converts input images into editable data [Purohit and Chauhan 2016]. Handwritten Character Recognition (HCR) is a type of OCR that recognizes handwritten text as opposed to Printed Character Recognition (PCR), which recognizes printed text [Agarwal *et al.* 2019]. OCR can be approached in two ways, namely segmentation-based and segmentation-free. Text images are segmented into individual characters before recognition in the first, while the latter avoids the segmentation step [Khaoula *et al.* 2013]. Segmentation-free OCR is more appropriate for handwritten text where the characters overlap, like cursive handwriting [Agarwal *et al.* 2019].

Depending on data acquisition, OCR can be categorized into offline and online recognition. Offline refers to recognizing data that is in rasterized format [Fogel *et al.* 2020], whereas online recognition recognizes data stored as a pen-tip ordered location sequence [Kumar *et al.* 2018a; Kang *et al.* 2020]. More research has been done for online handwriting systems due to the useful information they provide such as locus point and projection angles [Memon *et al.* 2020]. This dynamic information is useful to distinguish overlapping characters from each other [Priya *et al.* 2016]. Offline data is static and lacks this information.

Several approaches have been proposed to tackle OCR, however, it remains an open problem [Alonso *et al.* 2019]. Earlier methods used Template Matching Techniques [Memon *et al.* 2020]. The challenge is that it is complex to have generic templates for cursive text [Alonso *et al.* 2019]. Other proposed methods include Kernel Methods, Statistical Methods, Structure Pattern Recognition, and Artificial Neural Networks (ANN).

The most common techniques used under Kernel Methods include Support Vector Machines (SVMs) [Yang *et al.* 2005; Boukharouba and Bennia 2017], Kernel Principal Component Analysis and Kernel Fisher Discriminant Analysis [Verma and Ali 2012]. SVMs were used extensively before deep learning became prevalent [Memon *et al.* 2020]. k-Nearest Neighbor (kNN) [Chandio *et al.* 2018; PRADEEP *et al.* 2012; Kumar *et al.* 2018b; Lorigo and Govindaraju 2006; Liu *et al.* 2003; Boukharouba and Bennia 2017] and Hidden Markov Model (HMM) [Arica and Yarman-Vural 2001; Alma'adeed *et al.* 2002; Pechwitz and Maergner 2003; Alma'adeed *et al.* 2004; Cheriet 2008] are the most used techniques under Statistical Methods. ANN techniques include Multi Layer Preceptrons (MLP) [Liu and Suen 2009; Shamsher *et al.* 2007; Cireşan *et al.* 2010; Al-Jawfi 2009], Recurrent Neural Networks (RNN) [Su and Lu 2017; Graves *et al.* 2008; Chakraborty *et al.* 2016; Maalej *et al.* 2016; Graves and Schmidhuber 2009; Gupta *et al.* 2011] and Convolutional Neural Networks (CNN) [Yang *et al.* 2018; Sokar *et al.* 2018; Boufenar *et al.* 2018; Alizadehashraf and Roohi 2017; Ghasemi and Jadidinejad 2018; Lin *et al.* 2018]. RNN and CNN have reported remarkable achievements for OCR [Memon *et al.* 2020].

The techniques perform differently under the same metrics for different scripts. The reason is that the models exploit the style structure of characters to maximize performance, which differs between scripts [Memon *et al.* 2020]. Furthermore, performance depends on the quality of the dataset and the choice of features used [Awel and Abidi 2019]. The state-of-the-art models use CNNs in combination with other traditional procedures such as SVM and HMM [Memon *et al.* 2020].

OCR is a challenging pattern recognition problem [Purohit and Chauhan 2016]. Offline recognition has more challenges such as background clutters, illumination, camera angles, and character distortion. Most of the research published only focuses on one language or a subset thereof [Memon *et al.* 2020]. A generic system that can recognize different scripts and languages has not been built [Purohit and Chauhan 2016].

2.2.2 Writer Recognition

Writer recognition is a process that authenticates people on the basis of their handwriting. While handwriting recognition pays no attention to the writer's handwriting features, writer recognition uses these features to identify the writer. It is a useful biometric divided into writer identification and writer verification. Writer identification finds the writer of a document based on similarities to a stored reference list with documents of which the writers are known. Writer verification is a multimodal binary classifier that authenticates whether the same person wrote the documents in question [Siddiqi and Vincent 2010; Rehman *et al.* 2018].

The approaches to offline writer identification are classified into two: text-dependent and text-independent. The first requires the content of the written text to be identical (that is, it examines the same sequence of ASCII characters of the sample in question against the list of writers), while the latter can work with arbitrary texts (that is, it inspects the global handwriting style of the input sample against those of the writers). In addition, the first operates at word or character level while the latter on paragraph or line level. As a result, the text-dependent approach is more accurate but has limited value in practice, like in forensics. Text-independent writer identification is more general and difficult. This technique is similar to those used in signature verification [Sreeraj and Sumam 2011; Siddiqi and Vincent 2010; Rehman *et al.* 2018].

The categorization of writer recognition methods is similar to handwriting recognition. That is, we have statistical [Travieso *et al.* 2019; Christlein *et al.* 2017a; Kumar and Kaur 2017], structural [He and Schomaker 2017; Chahi *et al.* 2018; Pandey and Seeja 2018] and automatic model-based [Nguyen *et al.* 2019; Christlein *et al.* 2017b; Christlein and Maier 2018] feature extraction techniques [Rehman *et al.* 2018].

2.3 Generation

2.3.1 Background

Generative models form part of unsupervised learning frameworks that uncover the underlying structure of input data. Their aim is to imitate the data in a realistic manner such that it does not show that it was generated by a machine [Gonog and Zhou

2019]. Classic models are based on maximum likelihood, Markov chains and approximate inference [Pan *et al.* 2019]. Different models are hard to stack and combine due to difficulties in controlling the joint distribution and training. The models have different learning rates and may suffer from vanishing and exploding gradient problems. Deep Learning (DL) has proven itself to be the best for generative tasks [Oussidi and Elhassouny 2018]. The following section briefly discusses deep generative models.

Boltzmann Machines

A Boltzmann Machine [Fahlman *et al.* 1983] is a stochastic neural network and an energy-based model, a generative complement of Hopfield networks [Hinton 2007; Hopfield 2007]. The first proposed model has visible and hidden units made of undirected symmetric networks. It is not memory guided but instead learns the underlying structures of data. Theoretically, it can learn any probability distribution from a sample, however, it has not proven practicality in practice [Oussidi and Elhassouny 2018]. The algorithm learns very slowly with many layers. Restricted Boltzmann Machines (RBM) [Smolensky 1986], Deep Boltzmann Machines (DBM) [Salakhutdinov and Hinton 2009] and Deep Belief Networks (DBN) [Hinton 2009] are further works that improve on the initial model. RBM solved the tractability of joint distributions of the original Boltzmann Machine by creating independence within each of the two binary units, while having every visible node connected to every hidden node [Oussidi and Elhassouny 2018]. Training and sampling can be done using maximum likelihood [Tieleman 2008] and Markov Chain Monte Carlo methods [Andrieu *et al.* 2003] respectively. DBMs are a stack of RBMs, where each RBM is trained individually then fine-tuned by training the whole network using backpropagation [Hinton and Salakhutdinov 2012; Rumelhart *et al.* 1988]. DBMs can learn complex internal representations of data, but are slow and impractical for huge datasets. While DBMs are fully undirected, the top two layers in DBNs are directed towards the visible layer. Similar to DBMs, DBNs are a stack of RBMs. Training usually uses variations of the wake-sleep algorithm [Hinton *et al.* 1995]. These are not wholly generative models [Gonog and Zhou 2019].

Autoencoder

An autoencoder is an unsupervised feedforward nonrecurrent neural network that learns how to efficiently reduce (encode) the dimensionality of an input dataset by ignoring signal noise and decoding the reduced (encoded) representation to reconstruct a plausible representation of the original. The prior is called an encoder and the latter a decoder. Several variants, Sparse (SAE) [Ng and others 2011], Denoising (DAE) [Vincent *et al.* 2010], Contractive (CAE) [Vincent *et al.* 2008] and Variational (VAE) [Kingma and Welling 2013] autoencoders have been developed and restricted to induce useful properties in the reconstructed representation. Training can be done using backpropagation [Oussidi and Elhassouny 2018; Goodfellow *et al.* 2016]. SAE has a sparsity constraint on its loss function added by regularizing the mean square error cost function or k-sparse [Makhzani and Frey 2013]: manually zeroing all the code neurons except for the k strongest neurons that have the highest activation. DAE takes input that is partially corrupted and is trained to generate similar data to the undistorted distribution. CAE has an explicit regularizer added to its objective functions which forces it to be less sensitive to small variations in the input data.

VAEs are a combination of deep learning and Bayesian machine learning [Barber 2012] techniques, explicitly variational inference allowing them to encode the probability distribution of data in contrast to point encoding done by classic autoencoders [Zhang *et al.* 2019]. They have a continuous latent space by design, permitting random sampling and interpolation. As a result, they can generate new data. Contrarily, classical autoencoders do not guarantee continuity in their vector space, restricting them to only reproduce the input data [Weng 2018; Kiran *et al.* 2018; Oussidi and Elhassouny 2018]. MLP [Gardner and Dorling 1998], CNN [LeCun *et al.* 2015; Goodfellow *et al.* 2016] and RNN [LeCun *et al.* 2015] can be used to construct the encoder and decoder of VAEs [Zhang *et al.* 2019].

Generative Adversarial Networks

GANs [Goodfellow *et al.* 2014] are based on min-max game theory, posed as a zero-sum game where two networks, a discriminator (D) and a generator (G) compete to achieve Nash equilibrium [Gonog and Zhou 2019; Pan *et al.* 2019]. G takes stochastic noise and generates data while D classifies it as real or generated. G learns from D's feedback with no access to the real data input. The networks can be made up of MLP, RBM, CNN, etc. Training can be done using dropout algorithms and backpropagation, deeming approximate inference, and Markov chains unnecessary. GANs are actively being researched and have many variations mostly exploiting the structures of the data in their targeted use domain [Gonog and Zhou 2019]. Pan *et al.* [2019]; Hong *et al.* [2019] classified the variations of GANs based on optimization of architecture and objective function. Architectural optimization GANs are further divided into Convolution [Radford *et al.* 2015], Condition [Mirza and Osindero 2014; Odena *et al.* 2016; Chen *et al.* 2016], Hierarchy [Huang *et al.* 2017; Karras *et al.* 2017; Juefei-Xu *et al.* 2017] and Autoencoder [Dumoulin *et al.* 2016; Makhzani *et al.* 2015; Donahue *et al.* 2016] based GANs.

Deep Convolutional GANs (DCGAN) [Radford *et al.* 2015] improved the performance of the original GAN by replacing the Multi-Layer Perceptron (MLP) with Convolutional Neural Networks (CNN) which have been shown to outperform MLP. To have more control and not just randomly generate data, Mirza and Osindero [2014] introduced a conditional GAN (cGAN) which adds a conditional variable c as input to G and D to stabilize training and generate samples of a specific type. Variations of cGAN include InfoGAN [Chen *et al.* 2016] and Auxilary Classifier GAN (ACGAN) [Odena *et al.* 2016], where c is learned instead of given to the discriminator. Makhzani *et al.* [2015] proposed an Adversarial Autoencoder (AAE) which integrates adversarial networks and autoencoders. Bidirectional GANs (BiGANs) [Donahue *et al.* 2016] and Adversarially Learned Inference (ALI) [Dumoulin *et al.* 2016] improve on AAE's inability to learn the mapping from sample to latent space. Larsen *et al.* [2015] combined VAE and GAN and showed improved generation quality and reduced mode collapse. Brock *et al.* [2018] introduced BigGAN and showed that GANs benefit from large-scale training. The discussed are but a few variations that exist for GANs, Wang *et al.* [2019b]; Pan *et al.* [2019]; Hong *et al.* [2019]; Creswell *et al.* [2018]; Gui *et al.* [2020] provide descriptions of more GANs and applications in different fields.

The lack of a universal evaluation matric makes it difficult to measure the performance of GANs for different tasks. Inception Score (IS) [Salimans *et al.* 2016], Mode Score (MS) [Nowozin *et al.* 2016], Multi-scale Structural Similarity for Image Quality (MS-SSIM) [Wang *et al.* 2003], and Frèchet Inception Distance (FID) [Heusel *et al.* 2017] are some of the proposed metrics. IS measures the quality of generated samples but has mode collapse. MS can measure variation and visual quality. It is less sensitive to ground truth prior probability compared to IS. In contrast to IS, FID can discern intraclass mode dropping but is also prone to overfitting. Kernel Inception Distance (KID) [Bińkowski *et al.* 2018] solves this. MS-SSIM evaluates the similarity of images which is useful in evaluating mode collapse. Borji [2019] provide further analysis of other possible metrics to use. Researchers tend to use more than one of the metrics.

It has been shown difficult for GANs to reach Nash equilibrium [Arjovsky *et al.* 2017]. It can get wedged in a bad local minimum [Goodfellow 2016]. Other GAN challenges include counting the occurrences of objects, understanding the perspectives of 3D objects, and the global structure of the input. Mescheder *et al.* [2018] evaluate GAN training methods and their convergence.

Autoregressive networks

Autoregressive networks [Akaike 1969] learn the explicit distribution of the model structure imposes as opposed to GANs which learn the implicit distribution. Compared to GANs, autoregressive networks have better stability during training, can work for both continuous and discrete data, and offer a way to compute the likelihood. However, GANs are faster and can work without the provision of a probability density [Pan *et al.* 2019]. State-of-the-art variations include PixelRNN [van den Oord *et al.* 2016b], PixelCNN [van den Oord *et al.* 2016b], WaveNet [van den Oord *et al.* 2016a], PixelCNN++ [Salimans *et al.* 2017], and PixelSNAIL [Chen *et al.* 2018]. They are preferred for image completion due to their scalability and tractability which enable them to learn natural image distributions.

2.3.2 Related Work

Elanwar [2013] and Elarian *et al.* [2014] reviewed state-of-the-art handwriting synthesis techniques before the popularization of Deep Learning algorithms in handwriting generation. The following papers came after the reviews and have shown prominent enhancements from previously used methods. They have been divided into online and offline synthesis and highly focused on the Latin script, which the English and a majority of languages fall. Different techniques work better in different scripts as the scripts can be inherently discrete, cursive or both [Memon *et al.* 2020; Elarian *et al.* 2014]. There's been more work and success for online handwriting but people's online handwriting is substandard, hence the results as well. Offline synthesis can have more practical use than online synthesis.

Online

Graves [2013] proposed a novel idea that revolutionized handwriting synthesis. They approached handwriting synthesis as a sequence generation problem hence proposed the use of RNNs to approximate the probability distribution of the handwriting sequence. They added Long-Short Term Memory (LSTM) networks to increase the information remembered by the RNN. This, in turn, enabled the current values of the sequence and the hidden state of RNN to be able to predict the next probability density function of the next value of the sequence. A Mixture Density Networks (MDN) and Attention Mechanism were added to condition the network's predictions to a specific text sequence. The results produced look realistic. However, their model generates a hallucinated handwriting style rather than imitating a specified handwriting sample.

Kumar *et al.* [2018a] proposed training the top layer of the LSTM cell of the model proposed by Graves [2013] to imitate a sample handwriting style, but the model fell short to the limitations of RNNs and LSTMs. The model could not generate longer strings and retraining the top layer of the model was cumbersome, limiting the model for practical use. They did not add a huge improvement to Graves [2013]'s model.

A proposal from Ghosh *et al.* [2017] was to use a Deep Convolutional GAN (DCGAN). A third parameter that had the handwriting images with incorrect labels (ASCII character values) was added to the discriminator, in addition to the generated and real input images. As the discriminator learned to score them as fake, it learned not to only judge input images as real or fake, but to also match the character embedding. Reinforcement learning (RL) was used to join letters and form words. With this advancement, the generator learned to space characters and to make strokes from one letter to another better.

The model proposed by Ji and Chen [2019] is a modified GAN, where the discriminator has an integrated CNN-LSTM feature extraction and a Feedforward Neural Network classifier. The handwriting strokes are encoded following Path Signature Features (PSF). They used the generator proposed by Graves [2013]. The model generates handwritten texts which are neat and have miscellaneous styles and a uniform spatial distribution more than the model proposed by Graves [2013]. Turhan and Bilge [2018] combine CPPN (Compositional Pattern Producing Networks), VAE, and GAN models to generate high-resolution handwriting images. The new model (VAE/CPGAN) is shown to produce high-resolution images that outperform images generated from VAE [Kingma and Welling 2013; Salimans *et al.* 2014], VAE/GAN [Larsen *et al.* 2015], DCGAN [Ghosh *et al.* 2017; Radford *et al.* 2015] and CPPN-GAN-VAE [Ha 2016] evaluated using the Inception Score metric. Furthermore, VAE/CPGAN converges faster than the compared models.

The idea of Aksan *et al.* [2018] is to predict single pen positions, a model that is independent of the internal memory of the network to store style, in contrast to the attention mechanism model proposed by Graves [2013]. The proposal uses CVRNN to disentangle the input into two latent variables, style and content. This improved the control for style generation in Graves [2013]'s model better than Ji and Chen [2019]'s proposal. In a successive proposal, Aksan and Hilliges [2019] replaced the CVRNN with a Stochastic Temporal CNN (STCNN). The handwriting generation became more consistent.

Offline

Haines *et al.* [2016] use labeled segmented glyphs and structured texture synthesis to synthesize and imitate handwriting. They managed to produce realistic handwritten text that looks like that of the author. The challenge with the model is that it requires intensive human intervention to select glyphs and label the ligatures. An additional challenge is that the model cannot reproduce letters that are not in the input text.

Another novel idea which state-of-the-art models are built on was proposed by Alonso *et al.* [2019]. They use a modified GAN with an auxiliary network to assist in recognizing offline text. The generator is fed an encoded sequence of characters to be generated through a bidirectional LSTM recurrent layer. They integrate the generated images into the training data in contrast to all the papers above and below, except for Kang *et al.* [2020], and showed that it can increase recognition accuracy. The model cannot output varying sizes of words and it suffers from style collapse.

Fogel *et al.* [2020]'s work was inspired by Alonso *et al.* [2019]. Their model improves on the lack of varying length in words and images, and the need to annotate words at a character level. They proposed a semi-supervised fully convolutional handwriting text generator. A tweaked BigGAN was used. The handwriting images produced by this model were shown to be clearer, more versatile, and to have fewer artifacts under FID and GS metrics. This work, however, cannot generate characters with different receptive field widths.

Mayr *et al.* [2020] propose a fully automated spatial-temporal style transfer to imitate handwriting. They compute the skeleton of the input text using a proposed iterative knowledge transfer skeletonization algorithm. Afterward, they approximate the skeletonized sequence to an online sequence by converting the bitmap skeleton representations to strokes, and they obtain temporal information using maximum acceleration re-sampling and ordering. Graves [2013]'s model was used as the generator for handwriting synthesis. A modified *pix2pix* [Isola *et al.* 2017] was used to imitate ink and style of the original image from the online handwriting skeleton produced, hence producing realistic offline handwritten text. The produced text does not always look like that of the writer and is unrealistic when the skeletonization does not construct complete skeletons. Furthermore, the model has difficulties synthesizing punctuation marks.

Kang *et al.* [2020]'s proposal is similar to Haines *et al.* [2016] and Alonso *et al.* [2019]'s work. They proposed a GAN with the generative process conditioned with textual content and calligraphic style features. The model is non-recurrent to produce the final word image, removing the need for pen-tip position sequences.

2.3.3 Imitative Models and Deepfakes

StarGAN introduced by Choi *et al.* [2018] is used for image-to-image translation between domains. This can translate images with several attribute variations and has been shown to excel in facial feature transfer and expression synthesis than baseline models. It can learn from multiple dissimilar domain datasets. Pix2pixHD [Wang *et al.* 2018] uses semantic labels to synthesis photo-realistic high-resolution images, outperforming the state-of-the-art methods in pixel-wise correctness of semantic image segmentation. This work extends pix2pix [Isola *et al.* 2017] based on CGAN. Pix2pix is the first unified image-to-image translation method. GauGAN [Park *et al.* 2019] can create new landscapes in images with fewer artifacts than pix2pixHD. StarGAN is an unsupervised unimodal model whereas pix2pixHD is a supervised multimodal model. MUNIT [Huang *et al.* 2018] and Augmented CycleGAN [Almahairi *et al.* 2018] are unsupervised multimodal models, with MUNIT able to denote continuous output distributions.

StyleGAN [Karras *et al.* 2019a] is a GAN derived from style transfer literature. It automatically learns to separate high-level features (hair, freckles, facial pose) and to find stochastic variations in the images generated. It gives scale-specific control of the generation process and outperforms the state-of-the-art in interpolation quality and disentanglement of the variation's latent factors. StyleGAN2 [Karras *et al.* 2019b] improves on the architecture and training methods of StyleGAN resulting in a better quality of the synthesized images. RecycleGAN [Bansal *et al.* 2018] is a data-driven approach, combining spatial and temporal information for content transfer and style preservation. This GAN managed to capture subtle features (for example, dimples) in its synthesized clips. It's used for generating 'Deepfakes'.

A 3D-GAN, proposed by Wu *et al.* [2016] can generate 3D image models with logical lighting and reflections. It allows viewpoint shifts and texture and shape editing. Deep Recurrent Attentive Writer (DRAW) [Gregor *et al.* 2015] generates images in a sequential manner. It is based on VAEs and sequential attention mechanisms.

Texture synthesis can be broken into fine-grained synthesis (ground truth similarity) and coarse-grained synthesis (input-output similarity) [Wang *et al.* 2020]. Markovian GAN (MGAN) proposed by Hoang *et al.* [2018] generates stylized images and videos in real-time using captured Markovian patches. Spatial GAN [Jetchev *et al.* 2016] is the first proposed fully unsupervised texture synthesis model. Periodic Spatial GAN [Liu *et al.* 2018], inspired by SPGAN, can learn periodic textures, flexibly exploit texture information in noise space, and generate high-resolution textures of various sizes.

SRGAN [Ledig *et al.* 2017] improves the resolution of images with up to 4 times upscaling. The ESRGAN proposed by Wang *et al.* [2019a] is an improvement on SRGAN. They make advancements in the adversarial loss and perceptual loss of the network architecture. TGAN [Ding *et al.* 2019] synthesizes large high-resolution images from explored tensor structures.

Speech synthesis is the artificial generation of human speech. Both speech and handwriting generation are sequence problems [Graves 2013]. The WaveNet [van den Oord *et al.* 2016a] inspired by PixelCNN [van den Oord *et al.* 2016b] sounds more human-like than the state-of-the-art text-to-speech models. They chose CNN over RNN and LSTM due to their challenge of handling long time dependencies. van den Oord *et al.* [2017] builds upon the WaveNet model by combining it with Inverse autoregressive flows [Kingma *et al.* 2016]. They proposed a novel training method, Probability Density Distillation which is faster and more efficient.

Wang *et al.* [2017] proposed Tacotron, which is an end-to-end text-to-speech generative model where speech is synthesized at the frame level, making it faster compared to sample-level autoregressive methods. It is built on seq2seq [Sutskever *et al.* 2014] which has a post-processing net and an autoencoder with the decoder attention-based. A follow-up work, Tacotron2 [Shen *et al.* 2018], combines WaveNet and Tacotron, evaluating at 95% confidence intervals under MOS[Streijl *et al.* 2016]. A model that can clone voices given several samples was proposed by Arik *et al.* [2018]. The baseline of the proposal is upon Deep Voice 3 [Ping *et al.* 2017], which is fully convolutional and attention-based.

2.4 Datasets

Several benchmark datasets exist for offline handwriting recognition. To the best of our knowledge, there hasn't been one created for handwriting synthesis. However, the datasets that exist provide valuable attributes for handwriting generation. The IAPR has archived handwriting public datasets and their Ground Truths (GT). Google Dataset Search and Mendeley managed to index other public datasets apart from those at IAPR. Our focus was on offline handwritten words and characters by several writers.

The CVL-Database [Kleber *et al.* 2018] is an offline database produced for tasks such as writer identification and retrieval, and word-spotting. It has 311 writers (27 wrote 7 texts and 284 wrote 5 texts), each with a unique identifier. Of the 7 different texts they have, 6 are English and one is German. All the pages have the writer ID and text number. An XML file that has binds all single words and a GT-Viewer are also provided.

The IAM Handwriting Database [Marti and Bunke 2002] has 657 writes and 115320 words separated and labeled. It can be used for writer identification and verification. It has a similar style to the CVL-Dataset. Their data is obtained from scanned documents at 300dpi resolution. The IAM On-Line Handwriting Database (IAM-OnDB) [Liwicki and Bunke 2005] has 221 writers and 86272 words, of which 11059 are unique. A whiteboard was used to acquire the data.

Fiel *et al.* [2017] provides the Historical-WI dataset. 720 writers wrote 5 pages each, totaling to 3600 handwritten pages. They also released a training dataset of 394 writers writing 3 pages each. This was used for the ICDAR2017 competition on historical document writer identification. TriGraphSlant [Brink *et al.* 2011] is composed of 47 writers, each writing 4 pages. Page 1 and 2 are written using the writer's natural handwriting copying two district texts. To the best of their abilities, the writers slant their handwriting to the left and right on Page 3 and 4 respectively. The writers are Dutch, which is written in the Latin script.

IBM_UB_1 and IBM_UB_2 [Shivram *et al.* 2013] contains online and offline handwriting data. The offline data is scanned at 300dpi. IBM_UB_1 has 43 online writers and 41 offline writers contributing 6654 and 5934 pages respectively. The writers can be identified by their IDs. IBM_UB_2 has 200 French writers. GT is available at the line level while for IMB_UB_1 it is available at the word level. Both have an established correlation between offline and online handwritten documents. The TriGraphSlant, IAM dataset, and IBM_UB datasets are grey-scale images.

2.5 Conclusion

This chapter showed the trends, challenges, and open areas for research in handwriting generation and synthesis. Further, it provided the background information to which this research builds upon. Chapter 3 provides our Hypothesis, Research Questions and proposed Methodology to for this research.

Chapter 3

Research Methodology

3.1 Introduction

From Chapter 2 we saw the state of the art handwriting generative models, along with their limitations. In this chapter, we propose an approach that builds on the current models, with the aim that it will tackle the challenges faced by the earlier models. In Section 3.2 we provide our hypothesis followed by research questions in Section 3.3. The hypothesis and the research questions guide our experiments and evaluation that follow in Section 3.4 and Section 3.4.4 respectively. The Methodology section will go in-depth on the implementation of our model, while the Evaluation section will cover the metrics that will be used to test our results. Section 3.4.1 contains the datasets we will be using and Section 3.5 briefly reviews and concludes the chapter.

3.2 Hypothesis

By incorporating Triplet Loss into Variational Autoencoders, we can imitate a writer's handwriting style in a few shot paradigm. This can be accomplished using offline, online, and hybrid approaches.

3.3 Research Questions

Our research questions following the hypothesis are as follows:

1. Can we imitate a new writing style using zero and one-shot learning?

   We will investigate the ability of our model to synthesize handwriting styles when given a few samples of each writer during training. We also explore its ability to imitate handwriting styles it did not encounter during training.

2. How many samples of a new style are required before it can be imitated with comparable quality relative to the other styles?

   We investigate the amount of handwriting sample data the model needs in order to synthesize a new handwriting style with similar quality to the styles in the training set.

3. By providing the author information to the GANs proposed by Kang *et al.* [2020]; Fogel *et al.* [2020], are the GANs able to model multiple styles?

   We provide the writer style embedding to the above mentioned GANs to see if it will condition them to synthesize multiple witters' styles.

4. Are we able to inject the style embedding into the GAN approaches [Kang *et al.* 2020; Fogel *et al.* 2020] to allow for few-shot style imitation?

   We give the GANs proposed by the above mentioned the style embedding from the auto-encoder to see if they will be able to imitate a writer's style without having to retrain the whole GAN.

5. Do the quantitative evaluation metrics correlate to subject human assessments of the generated images?

   After we have done the human and quantitative evaluations, we will analyze the correlation of the results.

3.4 Methodology

3.4.1 Datasets

Several publicly available datasets are reviewed in Section 2.4. For the purpose of this research, we will be using the IAM On-Line Handwriting Database [Liwicki and Bunke 2005], and CVL Database [Kleber *et al.* 2018]. This is similar to Mayr *et al.* [2020] and will help us to compare results. The CVL dataset has also been used by Fogel *et al.* [2020] whose results we'll also be comparing in this research.

There are several text corpora that can be used for out-of-vocabulary words. For the purpose of this research, we'll be using the Leipzig 2016 English Wikipedia 1000000-word Corpus [Goldhahn *et al.* 2012], similar to Mayr *et al.* [2020]. Also, we remove the words that were already in the training set. To the resulting text corpus, we add the word 'supercalifragilisticexpialidocious', which was used to test for word-length generation by Fogel *et al.* [2020]. It is the firth longest word in the English language [Allen 2019]. We further add the five longest words under a hundred characters.

3.4.2 Proposed Models

We will use the following denotations for the models. Let K be the set of all writers during training, i.e., $K = \{(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)\}$, where X_i's are handwritten images and Y_i is the writer identity. The image annotations (handwritten text) is given by T. The generated images are denoted by $\bar{X}$. Let X' and Y' be writers who were not in the training set. That is, $Y' \not\subset Y$ and $Y' \cap Y = \emptyset$. A_y is a $d-$dimensional style feature vector that denotes the class embedding.

For temporal models we use $H = \{S_1, S_2, ..., S_t\}$ where S_i represents the i^{th} stroke.

$S_i = \{(x_1, y_1, s_1), (x_2, y_2, s_2), ..., (x_j, y_j, s_j)\}$ [1] where (x_l, y_l) is the l^{th} point of the stroke, and $s_l \in \{0, 1\}$ represents the up or down status of the pen. Let Q, P, z denote the encoder, decoder and latent space respectively. The generator and discriminator are denoted by G and D respectively.

Proposed Spatial Models (Offline Models)

Variational Auto-Encoder

We propose a VAE where the encoder takes the handwritten images, class embedding, and the content text to create a style embedding. The decoder takes the class embedding vector (A_y) and the text to be written. It uses this information to samples from the style embedding ($\mathbf{z}$) and generates handwritten images with the desired content and handwriting style. We minimize the Triplet Loss on the style embedding to force the same subjects to be collocated in the "style space". We give the text as input to force the model to only focus on the writer's style and not the text content (see Figure 3.1).

Generative Adverserial Model

We will use the offline handwriting GAN proposed by Fogel *et al.* [2020] to synthesize handwriting. Figure 3.2 depicts their entire architecture.

Hybrid Model

The proposed GAN is trained to generate one writer's handwriting style at a time. To train the GAN to generate different handwriting styles, we propose using the style embedding from the VAE as an input to the GAN model. This is in contrast to a style label that would be naturally used. However, this approach cannot generate new handwriting styles it didn't encounter during training. Figure 3.3 shows the proposed modification.

Proposed Temporal Models (Online Models)

Recurrent VAE

The VAE is the same as Section 3.4.2 but we replace the CNN in the encoder and decoder with LSTMs similar to Graves [2013]. This enables the model to learn order dependence in the stroke points. The model is illustrated in Figure 3.4.

GANs

Figure 3.5 illustrates the temporal GAN we will use for experimentation. The GAN adapts the generator proposed by Graves [2013].

[1] The temporal notation is adapted from Ji and Chen [2019].

Hybrid

We perform similar modifications as in Hybrid Model. The style embedding from the proposed CRVAE is fed into the generator during training.

Proposed Spatio-Temporal Models

Mayr *et al.* [2020]'s model cannot synthesize punctuation marks and ill handwritten samples. When their skeletonization process produces incomplete skeletons, the output looks unnatural. Kang *et al.* [2020] generates whole word images to combat the lack of spatio-temporal information in offline handwritten samples. However, this does not guarantee that the model will learn the character dependency. Our model aims to take the best of both worlds.

We design our model following the "Offline-to-Offline Handwriting Style Transfer Pipeline" proposed by Mayr *et al.* [2020]. Figure 3.6 illustrates the pipeline. It can be broken down into online approximation, text synthesis, and offline style transfer. We section the pipeline at the tasks we will modify.

Online approximation The online approximation maps offline to online handwritten data. An approximation is made since there is no database that annotates the mapping between online and offline data. The approximation starts off with a skeletonization phase, then they convert the bitmap skeleton to strokes, followed by a temporal resampling and ordering of the strokes. They proposed an iterative knowledge transfer to create the skeletons of the input data. This was chosen over CycleGAN [Zhu *et al.* 2017] which guarantees cycle consistency but not spatial consistency. The produced skeleton is then mapped to its temporal domain using maximum acceleration resampling, placing emphasis on curved strokes. The velocity is set to zero at line extremes, low on curves, and high on straight lines. This process attempts to mimic human writing dynamics. The resampled points are ordered from left to right. This is sometimes untrue for human handwriting but kept for consistency.

Text Synthesis We propose replacing Graves [2013]'s model with the best performing model from Proposed Temporal Models (Online Models). We also use Aksan and Hilliges [2019]'s model [2].

Offline style transfer Mayr *et al.* [2020] modifies pix2pix [Isola *et al.* 2017] to transfer the online strokes back to offline data by reproducing the ink and style of the input data. Figure 3.7 illustrates the modifications. The pix2pixHD model [Wang *et al.* 2018] has been proposed (see Figure 3.8). It tackles pix2pix's difficulty in producing high-resolution images and provides more stability in training the GAN. We propose similar modifications as Mayr *et al.* [2020] to the pix2pixHD. That is, taking the max-pooled

[2] They provide a pre-trained model. This is also suggested by Mayr *et al.* [2020]. To date, it has shown the best generative results for online handwriting generation.

outputs of the activation maps as the extracted global style of the input image. This is concatenated with the deepest layers of the network then fed into the pix2pixHD generator. As a result, the discriminator's loss function [3] becomes:

$$\begin{aligned}\mathcal{L}_{GAN}(G, D_k) = \mathbb{E}_{X,Y}&[\log D_k(X, Y, \hat{Y})] \\ &+ \mathbb{E}_{X,Y}[\log(1 - D_k(X, G(X, \hat{Y}), \hat{Y}))] \\ &+ \lambda_1 \mathbb{E}_{X,Y}[||Y - G(X, \hat{Y})||_1]\end{aligned} \tag{3.1}$$

for $k = 1, 2, 3$ where G is the generator, D_k is the k^{th} discriminator, Y is the image from which we want to extract the style from, $\hat{Y}$ is the style extracted from Y and X is the natural image to be rendered. λ_1 weighs the contribution of the style transfer network.

The full objective function remains unchanged, i.e.

$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k) \right) + \lambda \sum_{k=1,2,3} \mathcal{L}_{FM}(G, D_k) \right) \tag{3.2}$$

where the feature matching loss $\mathcal{L}_{FM}(G, D_k)$ is:

$$\mathcal{L}_{FM}(G, D_k) = \mathbb{E}_{X,Y} \sum_{i=1}^{T} \frac{1}{N_i} [||D_k^{(i)}(X, Y, \hat{Y}) - D_k^{(i)}(X, G(X, \hat{Y}), \hat{Y})||_1] \tag{3.3}$$

where T is the number of layers and N_i is the number of elements in each layer. λ in (3.2) weighs the significance of the two loss functions.

3.4.3 Experiments

We present a series of experiments, training, and tests that we will perform for this research.

Offline and Online Experiments

Spatial Models Training We start off by training the spatial models (Section 3.4.2) which we have deemed the easiest. We train the proposed GAN (Section 3.4.2) [4] to synthesize handwriting. The VAE model is trained to create the writer style embedding. After, we train the GAN with the style embedding from the VAE as input.

Spatial Models Comparison We compare the results produced by the three trained models from Spatial Models Training. We will visually inspect the results ourselves and use the quantitative metrics discussed in Section 3.4.4. The VAE and hybrid models are tested on various sample sizes of known and unknown handwriting styles.

[3] $\mathbb{E}_{X,Y} \triangleq \mathbb{E}_{(X,Y) \sim P_{data}(X,Y)}$

[4] The online GAN (Section 3.4.2) and offline GAN (Section 3.4.2) are trained first before the implementations. We implement and train the VAEs in parallel with the GAN training.

Temporal Models Training We follow the steps conducted in Section 3.4.4 but now implement them on the models proposed in Section 3.4.2.

Temporal Models Comparison The comparison methodology is the same as Spatial Models Comparison.

Few-shot learning We repeat the above VAE and hybrid model experiments varying the size of each handwriting style during training. By varying sizes of the training data we also achieve one-shot learning.

Spatio-Temporal Experiments

Hybrid Model We investigate using Mayr *et al.* [2020]'s proposal with the modifications discussed in Section 3.4.2. For the text synthesis step, we will take the best performing models from Offline and Online Experiments. Hence we train the text synthesizer using online and offline handwritten data.

Using Online Generative Models We will use Ji and Chen [2019]; Aksan and Hilliges [2019]'s text synthesis models for the text synthesis step in Mayr *et al.* [2020]'s model.

Comparison The comparison methodology is the same as Spatial Models Comparison.

Hybrid few-shot learning We repeat the Hybrid Model experiment with several models from Few-shot learning. From this experiment, we determine whether the hybrid model improves one-shot learning from the two individual models.

Human Acceptance

We perform the qualitative evaluation discussed on Section 3.4.4 using the results from the best performing model. The human qualitative results will be compared against the quantitative results to explore the correlation.

3.4.4 Evaluation

Several evaluation metrics have been proposed and used to evaluate and benchmark generative results. For this research, we will be using human perception for qualitative analysis and the numeric metrics for quantitative analysis. We need a combination of these in order to provide a robust evaluation of the performance of our model.

Qualitative Evaluation

Here we will be evaluating the visual aesthetics of the images. For this, we will be using people to look and rate the images. This is however not efficient and biased and makes it hard to compare and reproduce results. Metrics like Nearest Neighbours [Theis *et al.* 2015], Rating and Preference Judgment [Snell *et al.* 2017] and Rapid Scene Categorization [Oliva 2005] can be used for qualitative evaluation. However, earlier work in handwriting generation has not used these metrics. We also opt to use human vision since it allows us to customize the tests. We estimate we can get a minimum of a hundred people to do this.

Turing Test We assess the authenticity of the generated text. The Related Work to ours that uses people to judge gives them real and fake images to classify as real or fake. We strongly feel like this can be guesswork and does not fully show what people thought of the images. We propose using a Likert scale rather, with five options: highly synthetic, synthetic, neutral, realistic, highly realistic. This provides the confidence levels (CL) of each classification. Since the discriminator classifies images as real or fake, we also perform a human test where we give a person two images, and they should select which one is fake.

Mayr *et al.* [2020] uses a Google Form with each page showing an image asking a person to classify it as human written or machine-generated. The form has 32 tests, however, its arrangement makes it long. The way they approach this can definitely be improved. We propose building a simple website that is customized for this task. We will run three evaluations using: In-Vocabulary (IV) words, Out-of-Vocabulary (OOV) words, and long words.

For these evaluations there will be five real words and each of the experiments in Section 3.4.3 will also have five words (the people will not know) and their task is to classify all words that are real and fake stating their CL for each classification. When evaluating for 'long words' experiment, real words will not be there as we do not have real handwritten 'long words' data.

Writer Identification The Related Work in Section 2.3.2 reports their results using word images and not full sentences or paragraphs. This provides less data for text-independent writer identification (WI), favoring the use of a text-dependent WI. From the discussion in Section 2.2.2, we may only test for IV words. Since we have offline and online data, using human vision for this is simpler. Mayr *et al.* [2020] also uses human vision for this task. They use a Google Form with a real handwritten sentence, then place two words underneath, the user had to choose which word was written by the author of the sentence. They had 200 people each doing 64 classifications.

For our evaluation, we will use the proposed website. In contrast to Mayr *et al.* [2020] we perform the following experiments. For the first evaluation, we have three writers and their imitations, each writer and imitation has 5 images. People will be asked to classify the handwritten images by the writer. This is to see if the style is conserved. The second test is to see if the generated images are recognized as forged. We divide it into three: using the same text (IV), using IV different text, and using OOV different text. The proposed rating classification system in Turing Test is adhered

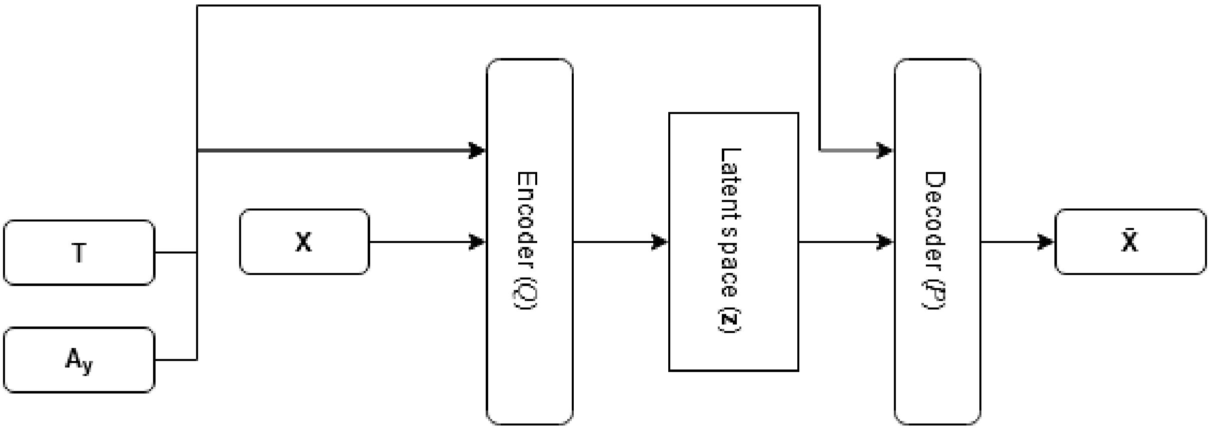


Figure 3.1: The conditional VAE (CVAE) takes the image (X), text (T), and writer (A_y) through an encoder $Q(z|X,T,A_y)$ which encodes it to the latent space. The decoder $Q(\bar{X}|T,A_y)$ samples z to generate new images ($\bar{X}$) with the text T using the handwriting style of A_y.

to. We argue however that handwriting forgery tends to have more artifacts in long handwritten text. That is, it is easier to imitate a word than a whole handwritten page. Hence, running the second test with as many images as possible should produce a better evaluation.

Quantitative Evaluation

Quantitative metrics summarize the quality of generated images using numerical scores [Borji 2019]. For this research, we will use the ones that have already been used in handwriting synthesis. However, other metrics may be used. We choose the following ones for comparative analysis. The Inception Score (IS) [Salimans *et al.* 2016], Frèchet Inception Distance (FID) [Heusel *et al.* 2017], Geometric Score (GS) [Khrulkov and Oseledets 2018] and Mean Average Precision (mAP) [Zhu 2004].

IS evaluates the quality of images and has been shown to correlate with human vision. FID measures the quality consistency of the generated images. GS compares the topology of the underlying images. In contrast to IS and FID, it is not constrained to visual data [Borji 2019]. The mAP is a standardized information retrieval metric that was used for the ICDAR2017 Competition on Historical Document Writer Identification [Fiel *et al.* 2017]. [Mayr *et al.* 2020] also uses this metric to evaluate writer imitation.

3.5 Conclusion

This chapter detailed the research proposal, starting from the hypothesis and research questions, to the experiments that we will run and the evaluations that we will conduct to answer the questions and prove or reject the hypothesis. In summary, the proposed model is hypothesized to work for both online and offline handwritten data, producing better results than the state of the art models measured through human vision and quantitative metrics. The following chapter details the plan to execute this proposal.

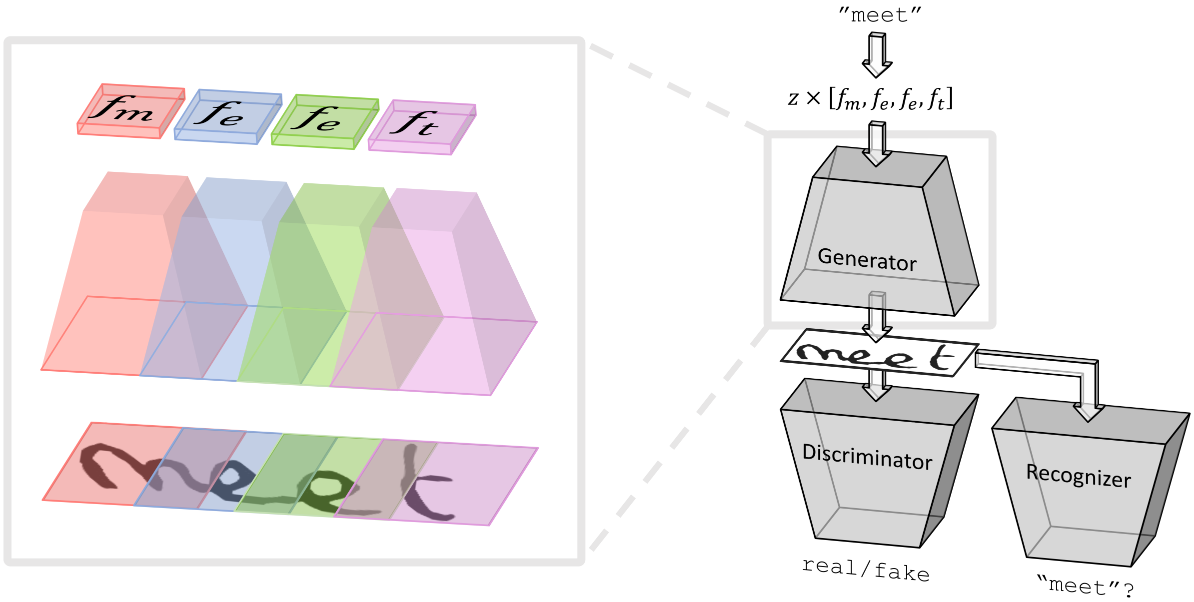


Figure 3.2: An overview of the model generating the word 'meet'. A noise vector z is concatenated with each character filter $f_{\star}$ and fed into G which generates an image that gets fed into D and an OCR/Recognizer (R). D inspects the authenticity of the generated image, while R assesses if the generated text is readable and the same as the input text. Adapted from [Fogel *et al.* 2020].

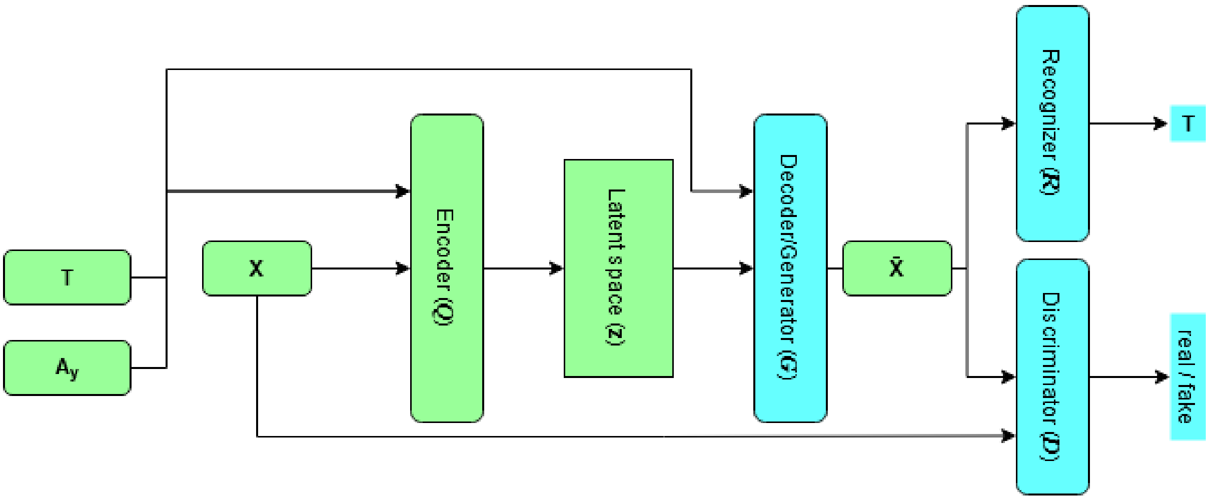
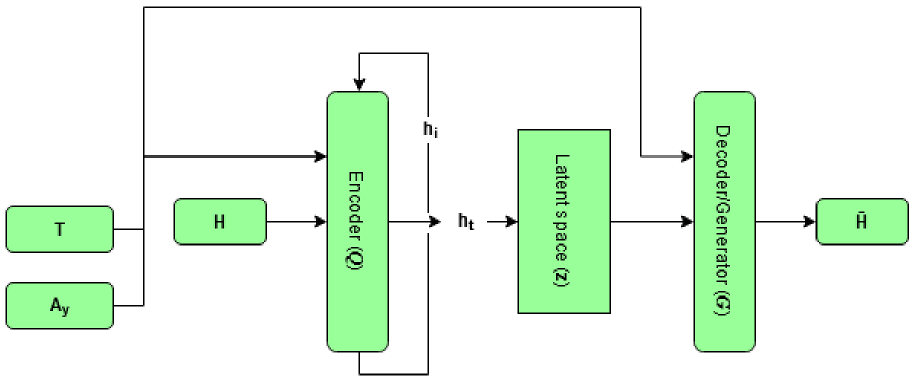
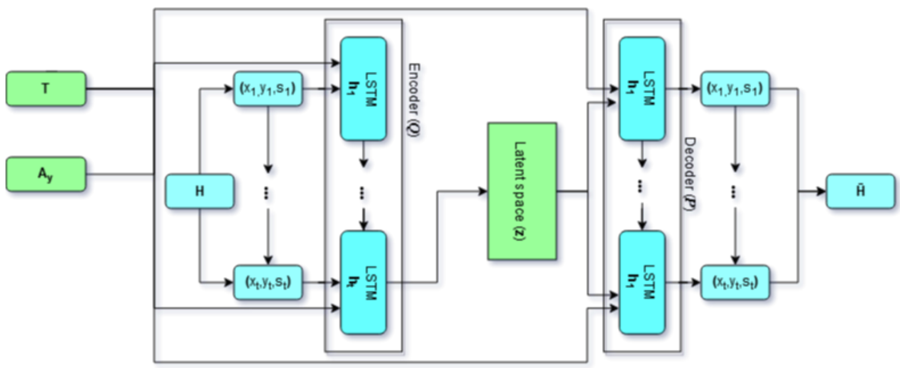


Figure 3.3: The generator from Fogel *et al.* [2020] is used as a decoder of the CVAE in Figure 3.1. Instead of the generator receiving random noise, it samples from the latent space (z). The rest of the model is left unchanged.

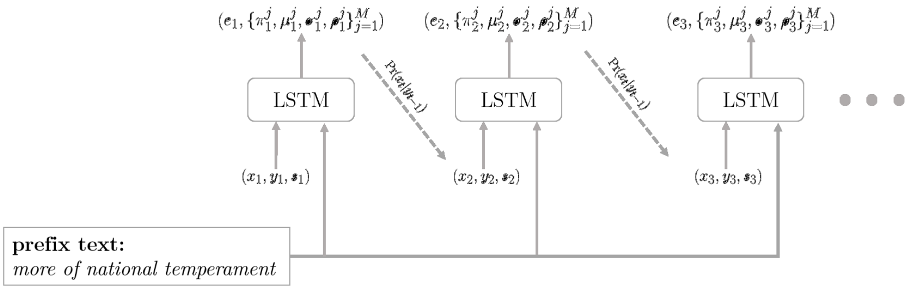


(a) Simplified CRVAE

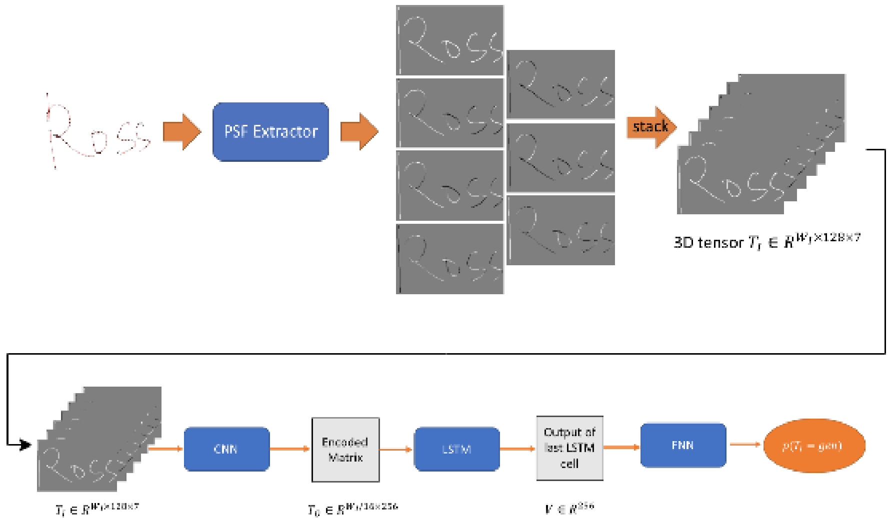


(b) CRVAE with expanded recurrent layers

Figure 3.4: The proposed conditional recurrent VAE. The difference with Figure 3.1 is the layers making up the encoder and decoder and the input. The encoder takes temporal information, i.e., the points in the sequence of strokes (S_i) using LSTMs. The decoder also uses LSTMs to sample the latent space and generate new strokes.



(a) Graves [2013]'s generator. Also used by Ji and Chen [2019].



(b) Discriminator model by Ji and Chen [2019]

Figure 3.5: The generator is made up of LSTMs which help it predict the stroke sequence. The discriminator takes strokes in binary format, then uses Path Signature Feature (PSF) to encode the geometrical and stroke order information. This is passed through a CNN which encodes the PSF into a 2D-matrix. The encoded matrix is passed to the LSTM sequentially. They use a Feedforward Neural Network (FNN) for classifying the input as real or fake. Adapted from Ji and Chen [2019].

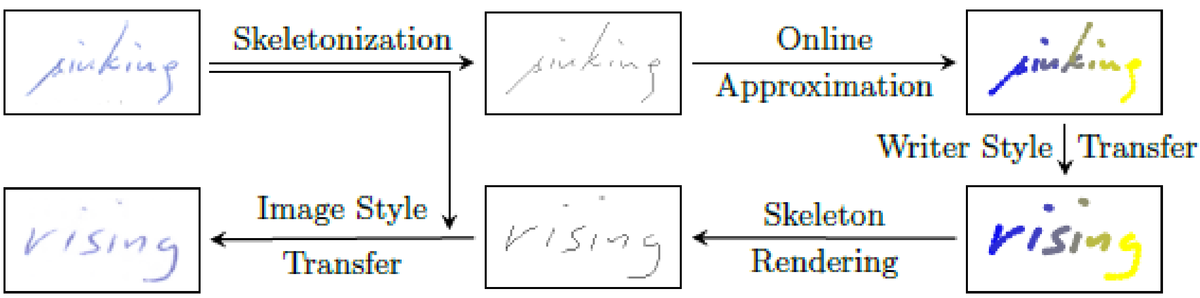


Figure 3.6: A Full Automated Offline-to-Offline Handwriting Style Transfer Pipeline. Adapted from [Mayr *et al.* 2020].

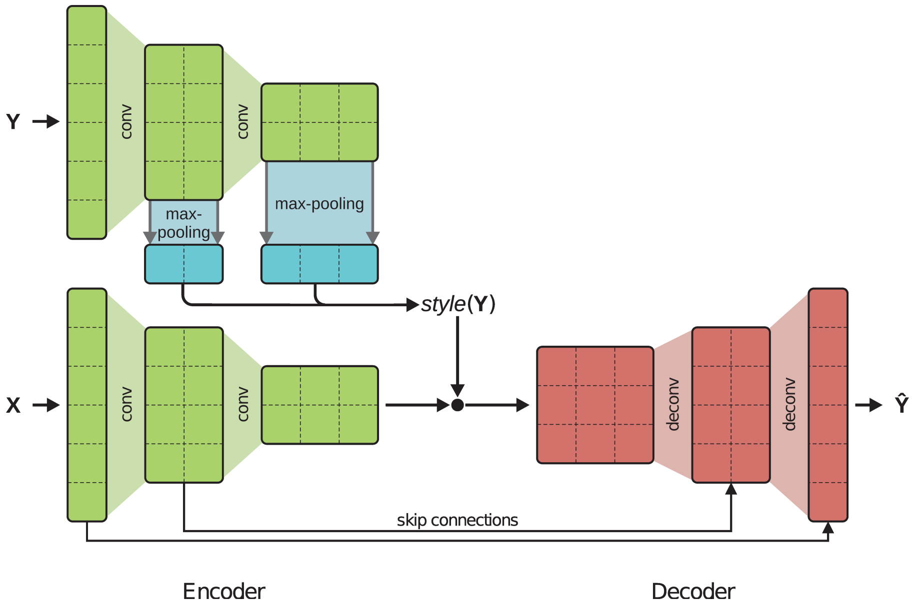


Figure 3.7: A style extraction network is added to the pix2pix generator network for conditional style transfer. Here, Y represents the offline image, X is the generated online skeleton and $\hat{Y}$ is the generated offline image. Adapted from [Fogel *et al.* 2020].

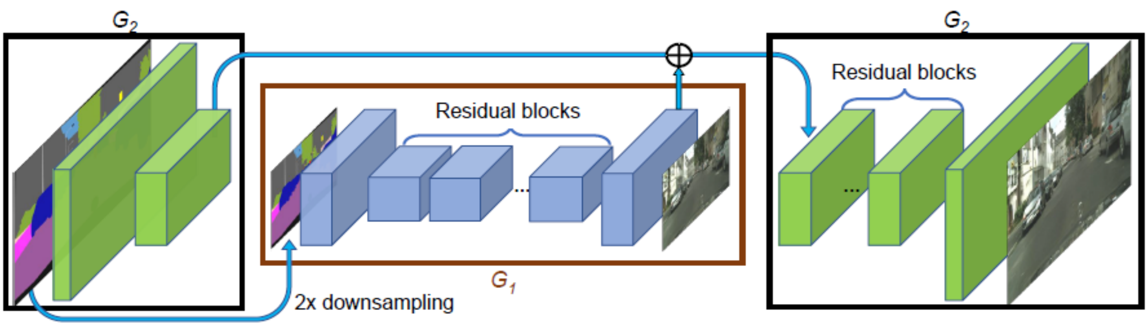


Figure 3.8: The pix2pixHD generator architecture. Adapted from [Wang *et al.* 2018].

Chapter 4

Research plan

4.1 Introduction

This chapter follows Chapter 3 where we proposed our research, including the Hypothesis, Methodology and Evaluation. Here we show the feasibility of the work given the time constraints. This is included in Section 4.2. We also evaluate the risks that may hinder this research from being successful in Section 4.3. Section 4.4 concludes our research plan.

4.2 Time plan

This section details our time plan for the proposed research. The second semester has 13 weeks. We plan our work based on this.

Week	Task [1]	Hours
July 17	Ji and Chen [2019] training [2]	15
24	Fogel *et al.* [2020] training [3]	12
31	Data collection, analysis and cleaning	15
Aug 07	VAE implementation	20
14	RVAE implementation	18
21	Offline hybrid implementation	30
28	Online hybrid implementation	25
Sep 4	Mid-term Vacation/Study/Research break	-
11	Online and offline experiments	30
18	pix2pixHD modification and training[4]	20
25	Spatio-temporal implementation [5]	30
Oct 02	Spatio-temporal experiments	30
09	Qualitative evaluation	30
16	Write-up	25
23	Peers and supervisor report review	10
30	Final draft and submission	15

4.3 Risks

A big evident risk is COVID-19 which has affected the academic year more than any past events in our generation. The Time plan will shift depending on the decisions of the university. We are currently at home where the network and computational power are a problem. However, the latter has a possible fix. We may run our experiments using the MSL cluster or other cloud services. Slow internet is an issue difficult to solve. However, there is a possibility that we will back to campus next semester. Supervision is also a problem with the current setup. It takes time before we can get hold of our supervisor. This causes significant delays.

Another risk is the lack of knowledge of how long each of the experiments will take to run. We have seen in Section 2.3.1 that GANs are difficult to train. We've made estimations, however, if they take longer than expected, we might not be able to do all the experiments. A possible solution is to use the study break between the third and fourth term to further the experiments. This provides an extra week to our time-plan. We estimated that we may get a minimum of a hundred people. This is a variable that's not entirely in our control. We are hoping that the university gives us permission to email students to participate. The university has over thirty thousand students. This will give us reviews from different faculties. However, if this does not work out, we rely

[1]The tasks can span over several weeks. The weeks serve to show the dates the tasks are due to be completed. When one task is complete, another is added regardless of the week.

[2]Ji and Chen [2019] provide source code on GitHub. The time amounts to us setting up the training.

[3]Fogel *et al.* [2020] provide source code on GitHub. The GANs are trained first, considering that they might take time to stabilize. We continue with the other tasks and leave them to train.

[4]The source code is available.

[5]Mayr *et al.* [2020] provide source code on GitHub.

on friends and colleagues who are over and above a hundred.

4.4 Conclusion

The chapter provided a time-based plan to follow on the proposed research. With the pandemic and uncertainty that befalls us, things might not go as planned. However, the plan is flexible to adapt.

Chapter 5

Conclusion

Handwriting synthesis presents itself as an interesting sequence generation problem which has seen several methods proposed to tackle it. The main focuses are on generating text images that look indistinguishable to text that is written by a human, reducing the amount of human assistance for the model, decreasing the amount of data required to train the model, imitating an author's handwriting from a sample not limited to predefined characters, generating out-of-vocabulary words and creating a model that is generic and scalable. Offline synthesis has to also synthesize background texture. To the best of our knowledge, there is no one model that solves all of these.

Benchmark databases have been established. Most state-of-the-art proposals use variations of GANs and VAEs, with offline synthesis trying to avoid character sequence generation due to the input being static and lacking temporal information. However, the generation of the final words does not guarantee the character to character conditioning that occurs in real handwriting. Mayr *et al.* [2020] proposed adding temporal information to the offline data to tackle this. However, the model does not always manage to add this information correctly. Inspired from this, we proposed a model that lies between the two approaches that exist, that is, character and full word generation.

We closely follow the approach by Mayr *et al.* [2020]. Our contribution is on the writer style transfer and offline style transfer. We proposed a hybrid model that uses the generated temporal information from their model with the writer's style extracted from the original offline images. This makes their model less dependent on their online handwriting approximation. For the offline style transfer, we modify pix2pixHD [Wang *et al.* 2018] which produces higher resolution results than the pix2pix [Isola *et al.* 2017] model they used. We further proposed VAE-GAN hybrids where the GANs are adapted from Fogel *et al.* [2020] and Ji and Chen [2019]. The style embedding from the VAEs are injected into the GANs to condition the GANs to multiple writer style transfer.

Another contribution lies in the evaluation of the results, to the best of our knowledge, it has not been used in handwriting synthesis or in other related generative domains.

References

[Agarwal *et al.* 2019] Megha Agarwal, Shalika, Vinam Tomar, and Priyanka Gupta. Handwritten character recognition using neural network and tensor flow. In *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, volume 8, pages 1445–1448, April 2019.

[Akaike 1969] Hirotugu Akaike. Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21(1):243–247, 1969.

[Aksan and Hilliges 2019] Emre Aksan and Otmar Hilliges. Stcn: Stochastic temporal convolutional networks. *arXiv preprint arXiv:1902.06568*, 2019.

[Aksan *et al.* 2018] Emre Aksan, Fabrizio Pece, and Otmar Hilliges. Deepwriting: Making digital ink editable via deep generative modeling. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018.

[Al-Jawfi 2009] Rashad Al-Jawfi. Handwriting arabic character recognition lenet using neural network. *Int. Arab J. Inf. Technol.*, 6:304–309, 2009.

[Alizadehashraf and Roohi 2017] B. Alizadehashraf and S. Roohi. Persian handwritten character recognition using convolutional neural network. In *2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pages 247–251, 2017.

[Allen 2019] Shundalyn Allen. *14 of the Longest Words in English*, May 2019. https://www.grammarly.com/blog/14-of-the-longest-words-in-english/. Accessed May 2020.

[Alma'adeed *et al.* 2002] S. Alma'adeed, C. Higgens, and D. Elliman. Recognition of off-line handwritten arabic words using hidden markov model approach. In *Object recognition supported by user interaction for service robots*, volume 3, pages 481–484 vol.3, 2002.

[Alma'adeed *et al.* 2004] Somaya Alma'adeed, Colin Higgins, and Dave Elliman. Off-line recognition of handwritten arabic words using multiple hidden markov models. *Knowledge-Based Systems*, 17(2):75 – 79, 2004. AI 2003, the Twenty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence.

[Almahairi *et al.* 2018] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. *Augmented CycleGAN: Learning Many-to-Many Mappings from Unpaired Data*, 2018.

[Alonso *et al.* 2019] E. Alonso, B. Moysset, and R. Messina. Adversarial generation of handwritten text images conditioned on sequences. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 481–486, Sep. 2019.

[Andrieu *et al.* 2003] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[Arica and Yarman-Vural 2001] N. Arica and F. T. Yarman-Vural. An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(2):216–233, 2001.

[Arik *et al.* 2018] Sercan O. Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. *Neural Voice Cloning with a Few Samples*, 2018.

[Arjovsky *et al.* 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[Awel and Abidi 2019] Muna Ahmed Awel and Ali Imam Abidi. Review on optical character recognition. *no. June*, pages 3666–3669, 2019.

[Bansal *et al.* 2018] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *ECCV*, 2018.

[Barber 2012] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[Beyerstein and Beyerstein 1992] Barry L Beyerstein and Dale F Beyerstein. *The write stuff: Evaluations of graphology, the study of handwriting analysis.* Prometheus Books, 1992.

[Bińkowski *et al.* 2018] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[Borji 2019] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, Feb 2019.

[Boufenar *et al.* 2018] Chaouki Boufenar, Adlen Kerboua, and Mohamed Batouche. Investigation on deep learning for off-line handwritten arabic character recognition. *Cognitive Systems Research*, 50:180 – 195, 2018.

[Boukharouba and Bennia 2017] Abdelhak Boukharouba and Abdelhak Bennia. Novel feature extraction technique for the recognition of handwritten digits. *Applied Computing and Informatics*, 13(1):19 – 26, 2017.

[Brink *et al.* 2011] A.A. Brink, R.M.J. Niels, R.A. van Batenburg, C.E. van den Heuvel, and L.R.B. Schomaker. *TriGraphSlant - benchmark set for writer identification - writers were asked to write in unnatural slant*, March 2011. https://doi.org/10.5281/zenodo.1195799. Accessed May 2020.

[Brock *et al.* 2018] Andrew Brock, Jeff Donahue, and Karen Simonyan. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*, 2018.

[Brown 1988] C. Marlin "Lin" Brown. Comparison of typing and handwriting in "two-finger typists". *Proceedings of the Human Factors Society Annual Meeting*, 32(5):381–385, 1988.

[Chahi *et al.* 2018] Abderrazak Chahi, Issam [El khadiri], Youssef [El merabet], Yassine Ruichek, and Raja Touahni. Block wise local binary count for off-line text-independent writer identification. *Expert Systems with Applications*, 93:1 – 14, 2018. http://www.sciencedirect.com/science/article/pii/S095741741730684X. Accessed May 2020.

[Chakraborty *et al.* 2016] Bappaditya Chakraborty, Partha Sarathi Mukherjee, and Ujjwal Bhattacharya. Bangla online handwriting recognition using recurrent neural network architecture. In *Proceedings of the tenth Indian conference on computer vision, graphics and image processing*, pages 1–8, 2016.

[Chandio *et al.* 2018] A. A. Chandio, M. Pickering, and K. Shafi. Character classification and recognition for urdu texts in natural scene images. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–6, 03 2018.

[Chen *et al.* 2016] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[Chen *et al.* 2018] XI Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. PixelSNAIL: An improved autoregressive generative model. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 864–872, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[Cheriet 2008] Mohamed Cheriet. Visual recognition of arabic handwriting: Challenges and new directions. In David Doermann and Stefan Jaeger, editors, *Arabic and Chinese Handwriting Recognition*, pages 1–21, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Choi *et al.* 2018] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[Christlein and Maier 2018] V. Christlein and A. Maier. Encoding cnn activations for writer recognition. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 169–174, 2018.

[Christlein *et al.* 2017a] Vincent Christlein, David Bernecker, Florian Hönig, Andreas Maier, and Elli Angelopoulou. Writer identification using gmm supervectors and exemplar-svms. *Pattern Recognition*, 63:258 – 267, 2017. http://www.sciencedirect.com/science/article/pii/S0031320316303211. Accessed May 2020.

[Christlein *et al.* 2017b] Vincent Christlein, Martin Gropp, Stefan Fiel, and Andreas Maier. Unsupervised feature learning for writer identification and writer retrieval. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Nov 2017. http://dx.doi.org/10.1109/ICDAR.2017.165. Accessed May 2020.

[Cireşan *et al.* 2010] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, Dec 2010.

[Creswell *et al.* 2018] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, Jan 2018.

[Ding *et al.* 2019] Zihan Ding, Xiao-Yang Liu, Miao Yin, and Linghe Kong. *TGAN: Deep Tensor Generative Adversarial Nets for Large Image Generation*, 2019.

[Donahue *et al.* 2016] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[Dumoulin *et al.* 2016] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. *Adversarially Learned Inference*, 2016.

[Elanwar 2013] Randa Elanwar. The state of the art in handwriting synthesis. In *2nd International Conference on New Paradigms in Electronics and information Technology (peit'013), Luxor, Egypt*, 12 2013.

[Elarian *et al.* 2014] Yousef Elarian, Radwan Abdel-Aal, Irfan Ahmad, Mohammad Parvez, and Abdelmalek Zidouri. Handwriting synthesis: Classifications and techniques. *INTERNATIONAL JOURNAL OF DOCUMENT ANALYSIS AND RECOGNITION*, 17, 09 2014.

[Fahlman *et al.* 1983] Scott Fahlman, Geoffrey Hinton, and Terrence Sejnowski. Massively parallel architectures for ai: Netl, thistle, and boltzmann machines. In *National Conference on Artificial Intelligence, AAAI*, pages 109–113, 01 1983.

[Fiel *et al.* 2017] Stefan Fiel, Florian Kleber, Markus Diem, Vincent Christlein, Georgios Louloudis, Nikos Stamatopoulos, and Basilis Gatos. *ScriptNet: ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI)*, August 2017. https://doi.org/10.5281/zenodo.1324999. Accessed May 2020.

[Fogel *et al.* 2020] Sharon Fogel, Hadar Averbuch-Elor, Sarel Cohen, Shai Mazor, and Roee Litman. *ScrabbleGAN: Semi-Supervised Varying Length Handwritten Text Generation*, 2020.

[Gardner and Dorling 1998] M.W Gardner and S.R Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14):2627 – 2636, 1998.

[Ghasemi and Jadidinejad 2018] S. Ghasemi and A. H. Jadidinejad. Persian text classification via character-level convolutional neural networks. In *2018 8th Conference of AI Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN)*, pages 1–6, 2018.

[Ghosh *et al.* 2017] Arna Ghosh, Biswarup Bhattacharya, and Somnath Basu Roy Chowdhury. Handwriting profiling using generative adversarial networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4927–4928. AAAI Press, 2017.

[Goldhahn *et al.* 2012] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2012/pdf/327_Paper.pdf. Accessed May 2020.

[Gonog and Zhou 2019] L. Gonog and Y. Zhou. A review: Generative adversarial networks. In *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 505–510, June 2019.

[Goodfellow *et al.* 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*, 2014.

[Goodfellow *et al.* 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[Goodfellow 2016] Ian Goodfellow. *NIPS 2016 Tutorial: Generative Adversarial Networks*, 2016.

[Graves and Schmidhuber 2009] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552, 2009.

[Graves *et al.* 2008] Alex Graves, Marcus Liwicki, Horst Bunke, Jürgen Schmidhuber, and Santiago Fernández. Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in neural information processing systems*, pages 577–584, 2008.

[Graves 2013] Alex Graves. *Generating Sequences With Recurrent Neural Networks*, 2013.

[Gregor *et al.* 2015] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. *DRAW: A Recurrent Neural Network For Image Generation*, 2015.

[Gui *et al.* 2020] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jie ping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *ArXiv*, abs/2001.06937, 2020.

[Gupta *et al.* 2011] Anshul Gupta, Manisha Srivastava, and Chitralekha Mahanta. Offline handwritten character recognition using neural network. *ICCAIE 2011 - 2011 IEEE Conference on Computer Applications and Industrial Electronics*, 12 2011.

[Ha 2016] David Ha. Generating large images from latent vectors. *blog.otoro.net*, 2016.

[Haines *et al.* 2016] Tom Haines, Oisin Aodha, and Gabriel Brostow. My text in your handwriting. *ACM Transactions on Graphics*, 35:1–18, 05 2016.

[He and Schomaker 2017] Sheng He and Lambert Schomaker. Writer identification using curvature-free features. *Pattern Recognition*, 63:451 – 464, 2017. http://www.sciencedirect.com/science/article/pii/S0031320316303053. Accessed May 2020.

[Heusel *et al.* 2017] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*, 2017.

[Hilton 1992] O. Hilton. *Scientific Examination of Questioned Documents, Revised Edition*. Forensic and Police Science Series. Taylor & Francis, 1992. https://books.google.co.za/books?id=-exqdS1GIHEC. Accessed May 2020.

[Hinton and Salakhutdinov 2012] Geoffrey E Hinton and Russ R Salakhutdinov. A better way to pretrain deep boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2447–2455, 2012.

[Hinton *et al.* 1995] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[Hinton 2007] Geoffrey E Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007.

[Hinton 2009] Geoffrey Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, January 2009.

[Hoang *et al.* 2018] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. MGAN: Training generative adversarial nets with multiple generators. In *International Conference on Learning Representations*, 2018.

[Hong *et al.* 2019] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. How generative adversarial networks and their variants work: An overview. *ACM Comput. Surv.*, 52(1), February 2019.

[Hopfield 2007] J. J. Hopfield. Hopfield network. *Scholarpedia*, 2(5):1977, 2007. revision #91363.

[Huang *et al.* 2017] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[Huang *et al.* 2018] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. *Lecture Notes in Computer Science*, page 179–196, 2018.

[Isola *et al.* 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[Jetchev *et al.* 2016] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. *Texture Synthesis with Spatial Generative Adversarial Networks*, 2016.

[Ji and Chen 2019] B. Ji and Tianyi Chen. Generative adversarial network for handwritten text. *ArXiv*, abs/1907.11845, 2019.

[Juefei-Xu *et al.* 2017] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. *Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking*, 2017.

[Kang *et al.* 2020] Lei Kang, Pau Riba, Yaxing Wang, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. *GANwriting: Content-Conditioned Generation of Styled Handwritten Word Images*, 2020.

[Karavanidou 2017] Eleni Karavanidou. Is handwriting relevant in the digital era? *Antistasis*, 7:153–164, 02 2017.

[Karras *et al.* 2017] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*, 2017.

[Karras *et al.* 2019a] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[Karras *et al.* 2019b] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. *Analyzing and Improving the Image Quality of StyleGAN*, 2019.

[Khaoula *et al.* 2013] Elagouni Khaoula, Christophe Garcia, Franck Mamalet, and Pascale Sébillot. Text recognition in multimedia documents: A study of two neural-based ocrs using and avoiding character segmentation. *International Journal on Document Analysis and Recognition (IJDAR)*, 17:1–13, 03 2013.

[Khrulkov and Oseledets 2018] Valentin Khrulkov and Ivan V. Oseledets. Geometry score: A method for comparing generative adversarial networks. In *ICML*, 2018.

[Kingma and Welling 2013] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*, 2013.

[Kingma *et al.* 2016] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. *Improving Variational Inference with Inverse Autoregressive Flow*, 2016.

[Kiran *et al.* 2018] B Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, 2018.

[Kleber *et al.* 2018] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. *CVL Database - An Off-line Database for Writer Retrieval, Writer Identification and Word Spotting*, November 2018. https://doi.org/10.5281/zenodo.1492267. Accessed May 2020.

[Kumar and Kaur 2017] Rohitash Kumar and Mandeep Kaur. A character based handwritten identification using neural network and svm. *International journal of scientific research in science, engineering and technology*, 3:120–124, 2017.

[Kumar *et al.* 2018a] K Manoj Kumar, Harish Kandala, and N Sudhakar Reddy. Synthesizing and imitating handwriting using deep recurrent neural networks and mixture density networks. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE, 2018.

[Kumar *et al.* 2018b] Munish Kumar, Simpel Jindal, M. Jindal, and Gurpreet Lehal. Improved recognition results of medieval handwritten gurmukhi manuscripts using boosting and bagging methodologies. *Neural Processing Letters*, 09 2018.

[Larsen *et al.* 2015] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. *Autoencoding beyond pixels using a learned similarity metric*, 2015.

[LeCun *et al.* 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[Ledig *et al.* 2017] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and et al. Photo-realistic single image super-resolution using a generative adversarial network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[Lin *et al.* 2018] Dazhen Lin, Fan Lin, Yanping Lv, Feipeng Cai, and Donglin Cao. Chinese character captcha recognition and performance estimation via deep neural network. *Neurocomputing*, 288:11 – 19, 2018. Learning System in Real-time Machine Vision.

[Liu and Suen 2009] Cheng-Lin Liu and Ching Y. Suen. A new benchmark on the recognition of handwritten bangla and farsi numeral characters. *Pattern Recognition*, 42(12):3287 – 3295, 2009. New Frontiers in Handwriting Recognition.

[Liu *et al.* 2003] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271 – 2285, 2003.

[Liu *et al.* 2018] Xiangyu Liu, Yunhong Wang, and Qingjie Liu. Psgan: A generative adversarial network for remote sensing image pan-sharpening. *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018.

[Liwicki and Bunke 2005] M. Liwicki and H. Bunke. Iam-ondb - an on-line english sentence database acquired from handwritten text on a whiteboard. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 956–961 Vol. 2, 2005.

[Lorigo and Govindaraju 2006] Liana Lorigo and Venu Govindaraju. Offline arabic handwriting recognition: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28:712 – 724, 06 2006.

[Maalej *et al.* 2016] Rania Maalej, Najiba Tagougui, and Monji Kherallah. Online arabic handwriting recognition with dropout applied in deep recurrent neural networks. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 417–421. IEEE, 2016.

[Makhzani and Frey 2013] Alireza Makhzani and Brendan Frey. *k-Sparse Autoencoders*, 2013.

[Makhzani *et al.* 2015] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. *Adversarial Autoencoders*, 2015.

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

[Mayr *et al.* 2020] Martin Mayr, Martin Stumpf, Anguelos Nikolaou, Mathias Seuret, Andreas Maier, and Vincent Christlein. *Spatio-Temporal Handwriting Imitation*, 2020.

[Memon *et al.* 2020] Jamshed Memon, Maira Sami, and Rizwan Ahmed Khan. *Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)*, 2020.

[Mescheder *et al.* 2018] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. *Which Training Methods for GANs do actually Converge?*, 2018.

[Mirza and Osindero 2014] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*, 2014.

[Ng and others 2011] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[Nguyen *et al.* 2019] Hung Nguyen, Cuong Nguyen, Takeya Ino, Bipin Indurkhya, and Masaki Nakagawa. Text-independent writer identification using convolutional neural network. *Pattern Recognition Letters*, 121:104 – 112, 2019. Graphonomics for e-citizens: e-health, e-society, e-education. http://www.sciencedirect.com/science/article/pii/S0167865518303180. Accessed May 2020.

[Nowozin *et al.* 2016] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

[Odena *et al.* 2016] Augustus Odena, Christopher Olah, and Jonathon Shlens. *Conditional Image Synthesis With Auxiliary Classifier GANs*, 2016.

[Oliva 2005] Aude Oliva. Chapter 41 - gist of the scene. In Laurent Itti, Geraint Rees, and John K. Tsotsos, editors, *Neurobiology of Attention*, pages 251 – 256. Academic Press, Burlington, 2005. http://www.sciencedirect.com/science/article/pii/B9780123757319500458. Accessed May 2020.

[Oussidi and Elhassouny 2018] A. Oussidi and A. Elhassouny. Deep generative models: Survey. In *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–8, 2018.

[Pan *et al.* 2019] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Zheng Yuhui. Recent progress on generative adversarial networks (gans): A survey. *IEEE Access*, PP:1–1, 03 2019.

[Pandey and Seeja 2018] Pallavi Pandey and KR Seeja. Forensic writer identification with projection profile representation of graphemes. In *Proceedings of First International Conference on Smart System, Innovations and Computing*, pages 129–136. Springer, 2018.

[Park *et al.* 2019] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[Pechwitz and Maergner 2003] M. Pechwitz and V. Maergner. Hmm based approach for handwritten arabic word recognition using the ifn/enit - database. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 890–894, 2003.

[Ping *et al.* 2017] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. *Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning*, 2017.

[PRADEEP *et al.* 2012] J. PRADEEP, E Srinivasan, and S Himavathi. Neural network based recognition system integrating feature extraction and classification for english handwritten. *International Journal of Engineering*, 25(2):99–106, 2012.

[Priya *et al.* 2016] A. Priya, S. Mishra, S. Raj, S. Mandal, and S. Datta. Online and offline character recognition: A survey. In *2016 International Conference on Communication and Signal Processing (ICCSP)*, pages 0967–0970, 2016.

[Purohit and Chauhan 2016] Ayush Purohit and Shardul Singh Chauhan. A literature survey on handwritten character recognition. *IJCSIT) International Journal of Computer Science and Information Technologies*, 7(1):1–5, 2016.

[Radford *et al.* 2015] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*, 2015.

[Rehman *et al.* 2018] Arshia Rehman, Saeeda Naz, and Muhammad Imran Razzak. Writer identification using machine learning approaches: a comprehensive review. *Multimedia Tools and Applications*, 78:10889–10931, 2018.

[Rumelhart *et al.* 1988] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988.

[Salakhutdinov and Hinton 2009] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009.

[Salimans *et al.* 2014] Tim Salimans, Diederik P. Kingma, and Max Welling. *Markov Chain Monte Carlo and Variational Inference: Bridging the Gap*, 2014.

[Salimans *et al.* 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[Salimans *et al.* 2017] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. *PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*, 2017.

[Shamsher *et al.* 2007] I Shamsher, Zaheer Ahmad, J Orakzai, and Awais Adnan. Ocr for printed urdu script using feed forward neural network. *the Proceedings of World Academy of Science, Engineering and Technology*, 23, 01 2007.

[Shen *et al.* 2018] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, and et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018.

[Shivram *et al.* 2013] A. Shivram, C. Ramaiah, S. Setlur, and V. Govindaraju. Ibm_ub_1: A dual mode unconstrained english handwriting dataset. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 13–17, Aug 2013.

[Siddiqi and Vincent 2010] Imran Siddiqi and Nicole Vincent. Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. *Pattern Recognition*, 43(11):3853 – 3865, 2010. http://www.sciencedirect.com/science/article/pii/S0031320310002438. Accessed May 2020.

[Smolensky 1986] Paul Smolensky. *Information processing in dynamical systems: Foundations of harmony theory*. Technical report, Colorado Univ at Boulder Dept of Computer Science, 1986.

[Snell *et al.* 2017] Jake Snell, Karl Ridgeway, Renjie Liao, Brett D. Roads, Michael C. Mozer, and Richard S. Zemel. Learning to generate images with perceptual similarity metrics. *2017 IEEE International Conference on Image Processing (ICIP)*, Sep 2017. http://dx.doi.org/10.1109/ICIP.2017.8297089. Accessed May 2020.

[Sokar *et al.* 2018] G. Sokar, E. E. Hemayed, and M. Rehan. A generic ocr using deep siamese convolution neural networks. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1238–1244, 2018.

[Sreeraj and Sumam 2011] M. Sreeraj and Mary Idicula Sumam. A survey on writer identification schemes. *International Journal of Computer Applications*, 26:23–33, 2011.

[Streijl *et al.* 2016] Robert C. Streijl, Stefan Winkler, and David S. Hands. Mean opinion score (mos) revisited: Methods and applications, limitations and alternatives. *Multimedia Syst.*, 22(2):213–227, March 2016.

[Su and Lu 2017] Bolan Su and Shijian Lu. Accurate recognition of words in scenes without character segmentation using recurrent neural network. *Pattern Recognition*, 63:397 – 405, 2017.

[Sutskever *et al.* 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[Theis *et al.* 2015] Lucas Theis, Aäron van den Oord, and Matthias Bethge. *A note on the evaluation of generative models*, 2015.

[Tieleman 2008] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071, 2008.

[Travieso *et al.* 2019] C. M. Travieso, J. L. Vásquez-Nuñez, and J. C. Briceño-Lobo. Analysis of the transformed contour in the writer identification. In *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 195–199, 2019.

[Turhan and Bilge 2018] C. G. Turhan and H. S. Bilge. Variational autoencoded compositional pattern generative adversarial network for handwritten super resolution image generation. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 564–568, Sep. 2018.

[van den Oord *et al.* 2016a] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. *WaveNet: A Generative Model for Raw Audio*, 2016.

[van den Oord *et al.* 2016b] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. *Pixel Recurrent Neural Networks*, 2016.

[van den Oord *et al.* 2017] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. *Parallel WaveNet: Fast High-Fidelity Speech Synthesis*, 2017.

[Verma and Ali 2012] Rohit Verma and Dr Jahid Ali. A-survey of feature extraction and classification techniques in ocr systems. *International Journal of Computer Applications & Information Technology*, 1(3):1–3, 2012.

[Vincent *et al.* 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.

[Vincent *et al.* 2010] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[Wang *et al.* 2003] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[Wang *et al.* 2017] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, and et al. Tacotron: Towards end-to-end speech synthesis. *Interspeech 2017*, Aug 2017.

[Wang *et al.* 2018] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[Wang *et al.* 2019a] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. *Computer Vision – ECCV 2018 Workshops*, page 63–79, 2019.

[Wang *et al.* 2019b] Zhengwei Wang, Qi She, and Tomas E. Ward. *Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy*, 2019.

[Wang *et al.* 2020] Lei Wang, Wei Chen, Wenjia Yang, Fangming Bi, and Fei Yu. A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access*, PP:1–1, 03 2020.

[Weng 2018] Lilian Weng. From autoencoder to beta-vae. *lilianweng.github.io/lil-log*, 2018.

[Wu *et al.* 2016] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[Yang *et al.* 2005] Lihua Yang, Ching Suen, T.D. Bui, and Ping Zhang. Discrimination of similar handwritten numerals based on invariant curvature features. *Pattern Recognition*, 38:947–963, 07 2005.

[Yang *et al.* 2018] H. Yang, L. Jin, and J. Sun. Recognition of chinese text in historical documents with page-level annotations. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 199–204, 2018.

[Zhang *et al.* 2019] Guijuan Zhang, Yang Liu, and Xiaoning Jin. A survey of autoencoder-based recommender systems. *Frontiers of Computer Science*, pages 1–21, 2019.

[Zhu *et al.* 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. http://dx.doi.org/10.1109/ICCV.2017.244. Accessed May 2020.

[Zhu 2004] Mu Zhu. *Recall, precision and average precision*. Technical report, University of Waterloo, Waterloo, 2004.