GANwriting: Content-Conditioned Generation of Styled Handwritten Word Images

Lei Kang*†, Pau Riba*, Yaxing Wang*, Marçal Rusiñol*, Alicia Fornés*, Mauricio Villegas†

*Computer Vision Center, Universitat Autònoma de Barcelona, Spain
{lkang, priba, yaxing, marcal, afornes}@cvc.uab.es
†omni:us, Berlin, Germany
{lei, mauricio}@omnius.com

Abstract. Although current image generation methods have reached impressive quality levels, they are still unable to produce plausible yet diverse images of handwritten words. On the contrary, when writing by hand, a great variability is observed across different writers, and even when analyzing words scribbled by the same individual, involuntary variations are conspicuous. In this work, we take a step closer to producing realistic and varied artificially rendered handwriting. We propose a novel method that is able to produce credible handwritten word images by conditioning the generative process with both calligraphic style features and textual content. Our generator is guided by three complementary learning objectives: to produce realistic images, to imitate a certain handwriting style and to convey a specific textual content. Our model is unconstrained to any predefined vocabulary, being able to render whatever input word. Given a sample writer, it is also able to mimic its calligraphic features in a few-shot setup. We significantly advance over prior art and demonstrate with qualitative, quantitative and human-based evaluations the realistic aspect of our synthetically produced images.

Keywords: Generative adversarial networks, style and content conditioning, handwritten word images.

1 Introduction

Few years after the conception of Generative Adversarial Networks (GANs) [12], we have witnessed an impressive progress on generating illusory plausible images. From the early low-resolution and hazy results, the quality of the artificially generated images has been notably enhanced. We are now able to synthetically produce high-resolution [5] artificial images that are indiscernible from real ones to the human observer [24]. In the original GAN architecture, inputs were randomly sampled from a latent space, so that it was hard to control which kind of images were being generated. The conception of conditional Generative Adversarial Networks (cGANs) [35] led to an important improvement. By allowing to condition the generative process on an input class label, the networks were then

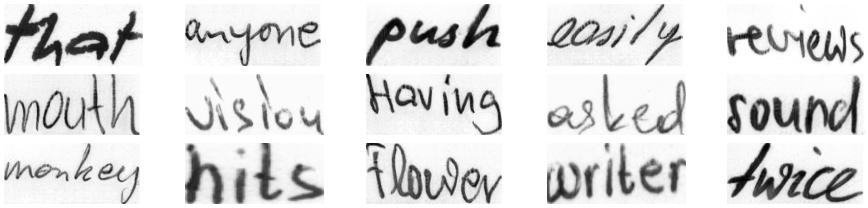


Fig. 1. Turing's test. Just five of the above words are real. Try to distinguish them from the artificially generated samples[1].

able to produce images from different given types [7]. However, such classes had to be predefined beforehand during the training stage and thus, it was impossible to produce images from other unseen classes during inference.

But generative networks have not exclusively been used to produce synthetic images. The generation of data that is sequential in nature has also been largely explored in the literature. Generative methods have been proposed to produce audio signals [9], natural language excerpts [45], video streams [42] or stroke sequences [13, 15, 11, 48] able to trace sketches, drawings or handwritten text. In all of those approaches, in order to generate sequential data, the use of Recurrent Neural Networks (RNNs) has been adopted.

Yet, for the specific case of generating handwritten text, one could also envisage the option of directly producing the final images instead of generating the stroke sequences needed to pencil a particular word. Such non-recurrent approach presents several benefits. First, the training procedure is more efficient since recurrencies are avoided and the inherent parallelism nature of convolutional networks is leveraged. Second, since the output is generated all at once, we avoid the difficulties of learning long-range dependencies as well as vanishing gradient problems. Finally, online training data (pen-tip location sequences), which is hard to obtain, is no longer needed.

Nevertheless, the different attempts to directly generate raw word images present an important drawback. Similarly to the case with cGANs, most of the proposed approaches are just able to condition the word image generation to a predefined set of words, limiting its practical use. For example [14] is specifically designed to generate isolated digits, while [6] is restricted to a handful of Chinese characters. To our best knowledge, the only exception to that is the approach by Alonso *et al.* [1]. In their work they propose a non-recurrent generative architecture conditioned to input content strings. By having such design, the generative process is not restricted to a particular predefined vocabulary, and could potentially generate any word. However, the produced results are not realistic, still exhibiting a poor quality, sometimes producing barely legible word images. Their proposed approach also suffers from the mode collapse problem, tending to produce images with a unique writing style. In this paper we present a non-recurrent generative architecture conditioned to textual content sequences,

[1] The real words are: "that", "vision", "asked", "hits" and "writer".

that is specially tailored to produce realistic handwritten word images, indistinguishable to humans. Real and generated images are actually difficult to tell apart, as shown in Fig. 1. In order to produce diverse styled word images, we propose to condition the generative process not only with textual content, but also with a specific writing style, defined by a latent set of calligraphic attributes.

Therefore, our approach[1] is able to artificially render realistic handwritten word images that match a certain textual content and that mimic some style features (text skew, slant, roundness, stroke width, ligatures, etc.) from an exemplar writer. To this end, we guide the learning process by three different learning objectives [36]. First, an adversarial discriminator ensures that the images are realistic and that its visual appearance is as closest as possible to real handwritten word images. Second, a style classifier guarantees that the provided calligraphic attributes, characterizing a particular handwriting style, are properly transferred to the generated word instances. Finally, a state-of-the-art sequence-to-sequence handwritten word recognizer [34] controls that the textual contents have been properly conveyed during the image generation. To summarize, the main contributions of the paper are the following:

- Our model conditions the handwritten word generative process both with calligraphic style features and textual content, producing varied samples indistinguishable by humans, surpassing the quality of the current state-of-the-art approaches.
- We introduce the use of three complementary learning objectives to guide different aspects of the generative process.
- We propose a character-based content conditioning that allows to generate any word, without being restricted to a specific vocabulary.
- We put forward a few-shot calligraphic style conditioning to avoid the mode collapse problem.

2 Related Work

The generation of realistic synthetic handwritten word images is a challenging task. To this day, the most convincing approaches involved an expensive manual intervention aimed at clipping individual characters or glyphs [43, 26, 28, 40, 16]. When such approaches were combined with appropriate rendering techniques including ligatures among strokes, textures and background blending, the obtained results were indeed impressive. Haines *et al.* [16] illustrated how such approaches could artificially generate indistinguishable manuscript excerpts as if they were written by Sir Arthur Conan Doyle, Abraham Lincoln or Frida Kahlo. Of course such manual intervention is extremely expensive, and in order to produce large volumes of manufactured images the use of truetype electronic fonts has also been explored [27, 23]. Although such approaches benefit from a greater scalability, the realism of the generated images clearly deteriorates.

[1] Our code is available at https://github.com/omni-us/research-GANwriting

With the advent of deep learning, the generation of handwritten text was approached differently. As shown in the seminal work by Alex Graves [13], given a reasonable amount of training data, an RNN could learn meaningful latent spaces that encode realistic writing styles and their variations, and then generate stroke sequences that trace a certain text string. However, such sequential approaches [13, 15, 11, 48] need temporal data, obtained by recording with a digital stylus pen real handwritten samples, stroke-by-stroke, in vector form.

Contrary to sequential approaches, non-recurrent generative methods have been proposed to directly produce images. Both variational auto-encoders [25] and GANs [12] were able to learn the MNIST manifold and generate artificial handwritten digit images in the original publications. With the emergence of cGANs [35], able to condition the generative process on an input image rather than a random noise vector, the adversarial-guided image-to-image translation problem started to rise. Image-to-image translation has since been applied to many different style transfer applications, as demonstrated in [21] with the *pix2pix* network. Since then, image translation approaches have been acquiring the ability to disentangle style attributes from the contents of the input images, producing better style transfer results [39, 37]. Geometry-aware synthesizing methods [47, 46] have been successfully applied on scene text images, but cursive words are not considered.

Concerning the generation of handwritten text, such approaches have been mainly used for synthesizing Chinese ideograms [30, 41, 6, 22, 44] and glyphs [2]. However, they are restricted to a predefined set of content classes. The incapability to generate out of vocabulary (OOV) text limits its practical application. Few works can actually deal with OOV words. First, in the work by Alonso *et al.* [1], the generation of handwritten word samples is conditioned by character sequences, but it suffers from the mode collapse problem, hindering the diversity of the generated images. Second, Fogel *et al.* [10] generate handwritten word by assembling the images generated by its characters, but the generated characters have the same receptive field width, which can make the generated words look unrealistic. Third, Mayr *et al.* [33] propose a conversion model to approximate online handwriting from offline data and then apply style transfer method to online data, so that offline handwritten text images could be generated by leveraging online handwriting synthesizer. However, this method highly depends on the performance of the conversion model and needs online data to train. Techniques like FUNIT [29], able to transfer unseen target styles to the content generated images could be beneficial for this limitation. In particular, the use of Adaptive Instance Normalization (AdaIN) layers, proposed in [18], shall allow to align both textual content and style attributes within the generative process.

Summarizing, state-of-the-art generative methods are still unable to produce plausible yet diverse images of whatever handwritten word automatically. In this paper we propose to condition a generative model for handwritten words with unconstrained text sequences and stylistic typographic attributes, so that we are able to generate any word with a great diversity over the produced results.

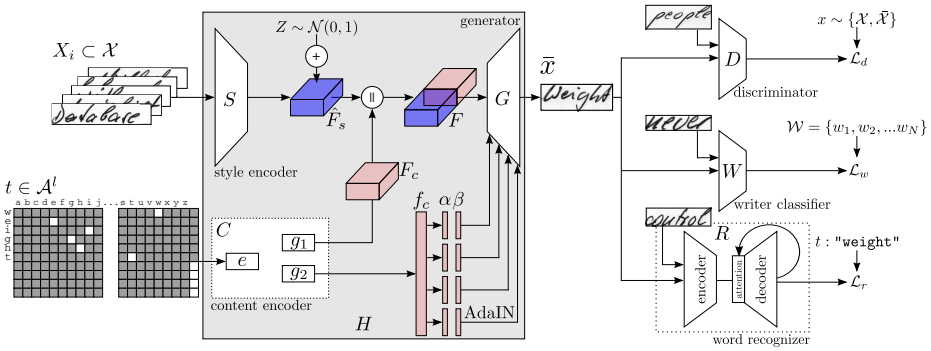


Fig. 2. Architecture of the proposed handwriting generation model.

3 Conditioned Handwritten Generation

3.1 Problem Formulation

Let $\{\mathcal{X}, \mathcal{Y}, \mathcal{W}\}$ be a multi-writer handwritten word dataset, containing grayscale word images $\mathcal{X}$, their corresponding transcription strings $\mathcal{Y}$ and their writer identifiers $\mathcal{W} = \{w_i\}_{i=1}^{N}$. Let $X_i = \{x_{w_i,j}\}_{j=1}^{K} \subset \mathcal{X}$ be a subset of K randomly sampled handwritten word images from the same given writer $w_i \in \mathcal{W}$. Let $\mathcal{A}$ be the alphabet containing the allowed characters (letters, digits, punctuation signs, etc.), $\mathcal{A}^l$ being all the possible text strings with length l. Given a set of images X_i as a few-shot example of the calligraphic style attributes for writer w_i on the one hand, and given a textual content provided by any text string $t \in \mathcal{A}^l$ on the other hand; the proposed generative model has the ability to combine both sources of information. It has the objective to yield a handwritten word image having textual content equal to t and sharing calligraphic style attributes with writer w_i. Following this formulation, the generative model H is defined as

$$\bar{x} = H\left(t, X_i\right) = H\left(t, \{x_1, \ldots, x_K\}\right), \tag{1}$$

where $\bar{x}$ is the artificially generated handwritten word image with the desired properties. Moreover, we denote $\bar{\mathcal{X}}$ as the output distribution of the generative network H.

The proposed architecture is divided in two main components. The generative network produces human-readable images conditioned to the combination of calligraphic style and textual content information. The second component are the learning objectives which guide the generative process towards producing images that look realistic; exhibiting a particular calligraphic style attributes; and having a specific textual content. Fig. 2 gives an overview of our model.

3.2 Generative Network

The proposed generative architecture H consists of a calligraphic style encoder S, a textual content encoder C and a conditioned image generator G. The overall

calligraphic style of input images X_i is disentangled from their individual textual contents, whereas the string t provides the desired content.

Calligraphic style encoding. Given the set $X_i \subset \mathcal{X}$ of $K = 15$ word images from the same writer w_i, the style encoder aims at extracting the calligraphic style attributes, *i.e.* slant, glyph shapes, stroke width, character roundness, ligatures etc. from the provided input samples. Specifically, our proposed network S learns a style latent space mapping, in which the obtained style representations $F_s = S(X_i)$ are disentangled from the actual textual contents of the images X_i. The VGG-19-BN [38] architecture is used as the backbone of S. In order to process the input image set X_i, all the images are resized to have the same height h, padded to meet a maximum width w and concatenated channel-wise to end up with a single tensor $h \times w \times K$. If we ask a human to write the same word several times, slight involuntary variations appear. In order to imitate this phenomenon, randomly choosing permutations of the subset X_i will already produce such characteristic fluctuations. In addition, an additive noise $Z \sim \mathcal{N}(0, 1)$ is applied to the output latent space to obtain a subtly distorted feature representation $\hat{F}_s = F_s + Z$.

Textual content encoding. The textual content network C is devoted to produce an encoding of the given text string t that we want to artificially write. The proposed architecture outputs content features at two different levels. Low-level features encode the different characters that form a word and their spatial position within the string. A subsequent broader representation aims at guiding the whole word consistency. Formally, let $t \in \mathcal{A}^l$ be the input text string, character sequences shorter than l are padded with the empty symbol ε. Let us define a character-wise embedding function $e\colon \mathcal{A} \to \mathbb{R}^n$. The first step of the content encoding stage embeds with a linear layer each character $c \in t$, represented by a one-hot vector, into a character-wise latent space. Then, the architecture is divided into two branches.

Character-wise encoding: Let $g_1\colon \mathbb{R}^n \to \mathbb{R}^m$ be a Multi-Layer Perceptron (MLP). Each embedded character $e(c)$ is processed individually by g_1 and their results are later stacked together. In order to combine such representation with style features, we have to ensure that the content feature map meets the shape of $\hat{F}_s$. Each character embedding is repeated multiple times horizontally to coarsely align the content features with the visual ones extracted from the style network, and the tensor is finally vertically expanded. The two feature representations are concatenated to be fed to the generator $F = [\hat{F}_s \parallel F_c]$. Such a character-wise encoding enables the network to produce OOV words, *i.e.* words that have never been seen during training.

Global string encoding: Let $g_2\colon \mathbb{R}^{l \cdot n} \to \mathbb{R}^{2p \cdot q}$ be another MLP aimed at obtaining a much broader and global string representation. The character embeddings $e(c)$ are concatenated into a large one-dimensional vector of size $l \cdot n$ that is then processed by g_2. Such global representation vector f_c will be then injected into the generator splitted into p pairs of parameters.

Both functions $g_1(\cdot)$ and $g_2(\cdot)$ make use of three fully-connected layers with ReLU activation functions and batch normalization [20].

Generator. Let F be the combination of the calligraphic style attributes and the textual content information character-wise; and f_c the global textual encoding. The generator G is composed of two residual blocks [19] using the AdaIN as the normalization layer. Then, four convolutional modules with nearest neighbor upsampling and a final tanh activation layer generates the output image $\bar{x}$. AdaIN is formally defined as

$$\text{AdaIN}\left(z, \alpha, \beta\right) = \alpha \left(\frac{z - \mu\left(z\right)}{\sigma\left(z\right)} \right) + \beta, \tag{2}$$

where $z \in F$, μ and σ are the channel-wise mean and standard deviations. The global content information is injected four times ($p = 4$) during the generative process by the AdaIN layers. Their parameters α and β are obtained by splitting f_c in four pairs. Hence, the generative network is defined as

$$\bar{x} = H\left(t, X_i\right) = G\left(C\left(t\right), S\left(X_i\right)\right) = G\left(g_1\left(\hat{t}\right), g_2\left(\hat{t}\right), S\left(X_i\right)\right), \tag{3}$$

where $\hat{t} = [e(c); \forall c \in t]$ is the encoding of the string t character by character.

3.3 Learning Objectives

We propose to combine three complementary learning objectives: a discriminative loss, a style classification loss and a textual content loss. Each one of these losses aim at enforcing different properties of the desired generated image $\bar{x}$.

Discriminative Loss. Following the paradigm of GANs [12], we make use of a discriminative model D to estimate the probability that samples come from a real source, *i.e.* training data $\mathcal{X}$, or belong to the artificially generated distribution $\bar{\mathcal{X}}$. Taking the generative network H and the discriminator D, this setting corresponds to a min max optimization problem. The proposed discriminator D starts with a convolutional layer, followed by six residual blocks with LeakyReLU activations and average poolings. A final binary classification layer is used to discern between fake and real images. Thus, the discriminative loss only controls that the general visual appearance of the generated image looks realistic. However, it does not take into consideration neither the calligraphic styles nor the textual contents. This loss is formally defined as

$$\mathcal{L}_d\left(H, D\right) = \mathbb{E}_{x \sim \mathcal{X}}\left[\log\left(D\left(x\right)\right)\right] + \mathbb{E}_{\bar{x} \sim \bar{\mathcal{X}}}\left[\log\left(1 - D\left(\bar{x}\right)\right)\right]. \tag{4}$$

Style Loss. When generating realistic handwritten word images, encoding information related to calligraphic styles not only provides diversity on the generated samples, but also prevents the mode collapse problem. Calligraphy is a strong identifier of different writers. In that sense, the proposed style loss guides the generative network H to generate samples conditioned to a particular writing style by means of a writer classifier W. Given a handwritten word image, W tries to identify the writer $w_i \in \mathcal{W}$ who produced it. The writer classifier W follows the same architecture of the discriminator D with a final classification MLP with the amount of writers in our training dataset. The classifier W is only

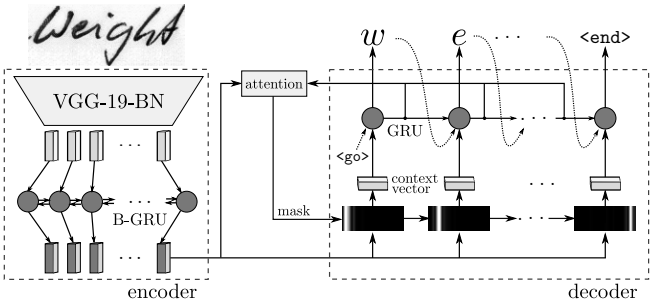


Fig. 3. Architecture of the attention-based sequence-to-sequence handwritten word recognizer R.

optimized with real samples drawn from $\mathcal{X}$, but it is used to guide the generation of the synthetic ones. We use the cross entropy loss, formally defined as

$$\mathcal{L}_w\left(H, W\right) = -\mathbb{E}_{x\sim\{\mathcal{X},\bar{\mathcal{X}}\}}\left[\sum_{i=1}^{|\mathcal{W}|} w_i \log\left(\hat{w}_i\right)\right], \tag{5}$$

where $\hat{w} = W(x)$ is the predicted probability distribution over writers in $\mathcal{W}$ and w_i the real writer distribution. Generated samples should be classified as the writer w_i used to construct the input style conditioning image set X_i.

Content Loss. A final handwritten word recognizer network R is used to guide our generator towards producing synthetic word images with a specific textual content. We implemented a state-of-the-art sequence-to-sequence model [34] for handwritten word recognition to examine whether the produced images $\bar{x}$ are actually decoded as the string t. The recognizer, depicted in Fig. 3, consists of an encoder and a decoder coupled with an attention mechanism. Handwritten word images are processed by the encoder and high-level feature representations are obtained. A VGG-19-BN [38] architecture followed by a two-layered Bi-directional Gated Recurrent Unit (B-GRU) [8] is used as the encoder network. The decoder is a one-directional RNN that outputs character by character predictions at each time step. The attention mechanism dynamically aligns context features from each time step of the decoder with high-level features from the encoder, hopefully corresponding to the next character to decode. The Kullback-Leibler divergence loss is used as the recognition loss at each time step. This is formally defined as

$$\mathcal{L}_r\left(H, R\right) = -\mathbb{E}_{x\sim\{\mathcal{X},\bar{\mathcal{X}}\}}\left[\sum_{i=0}^{l}\sum_{j=0}^{|\mathcal{A}|} t_{i,j} \log\left(\frac{t_{i,j}}{\hat{t}_{i,j}}\right)\right], \tag{6}$$

where $\hat{t} = R(x)$; $\hat{t}_i$ being the i-th decoded character probability distribution by the word recognizer, $\hat{t}_{i,j}$being the probability of j-th symbol in $\mathcal{A}$ for $\hat{t}_i$, and $t_{i,j}$ being the real probability corresponding to $\hat{t}_{i,j}$. The empty symbol ε is ignored in the loss computation; t_i denotes the i-th character on the input text t.

Algorithm 1 Training algorithm for the proposed model.

Input: Input data $\{\mathcal{X}, \mathcal{Y}, \mathcal{W}\}$; alphabet $\mathcal{A}$; max training iterations T
Output: Networks parameters $\{\Theta_H, \Theta_D, \Theta_W, \Theta_R\}$.

1: **repeat**
2: Get style and content mini-batches $\{X_i, w_i\}_{i=1}^{N_B}$ and $\{t^i\}_{i=1}^{N_B}$
3: $\mathcal{L}_d \leftarrow$ Eq. 4 ▷ Real and generated samples $x \sim \{\mathcal{X}, \bar{\mathcal{X}}\}$
4: $\mathcal{L}_{w,r} \leftarrow$ Eq. 5 + Eq. 6 ▷ Real samples $x \sim \mathcal{X}$
5: $\Theta_D \leftarrow \Theta_D + \Gamma(\nabla_{\Theta_D} \mathcal{L}_d)$
6: $\Theta_{W,R} \leftarrow \Theta_{W,R} - \Gamma(\nabla_{\Theta_{W,R}} \mathcal{L}_{w,d})$
7: $\mathcal{L} \leftarrow$ Eq. 7 ▷ Generated samples $x \sim \bar{\mathcal{X}}$
8: $\Theta_H \leftarrow \Theta_H - \Gamma(\nabla_{\Theta_H} \mathcal{L})$
9: **until** Max training iterations T

3.4 End-to-end Training

Overall, the whole architecture is trained end to end with the combination of the three proposed loss functions

$$\mathcal{L}(H, D, W, R) = \mathcal{L}_d(H, D) + \mathcal{L}_w(H, W) + \mathcal{L}_r(H, R), \tag{7}$$

$$\min_{H,W,R} \max_{D} \mathcal{L}(H, D, W, R). \tag{8}$$

Algorithm 1 presents the training strategy that has been followed in this work. $\Gamma(\cdot)$ denotes the optimizer function. Note that the parameter optimization is performed in two steps. First, the discriminative loss is computed using both real and generated samples (line 3). The style and content losses are computed by just providing real data (line 4). Even though W and D are optimized using only real data and, therefore, they could be pre-trained independently from the generative network H, we obtained better results by initializing all the networks from scratch and jointly training them altogether. The network parameters Θ_D are optimized by gradient ascent following the GAN paradigm whereas the parameters Θ_W and Θ_R are optimized by gradient descent. Finally, the overall generator loss is computed following Equation 7 where only the generator parameters Θ_H are optimized (line 8).

4 Experiments

To carry out the different experiments, we have used a subset of the IAM corpus [32] as our multi-writer handwritten dataset $\{\mathcal{X}, \mathcal{Y}, \mathcal{W}\}$. It consists of $62,857$ handwritten word snippets, written by 500 different individuals. Each word image has its associated writer and transcription metadata. A test subset of 160 writers has been kept apart during training to check whether the generative model is able to cope with unseen calligraphic styles. We have also used a subset of $22,500$ unique English words from the Brown [3] corpus as the source of strings for the content input. A test set of 400 unique words, disjoint from the

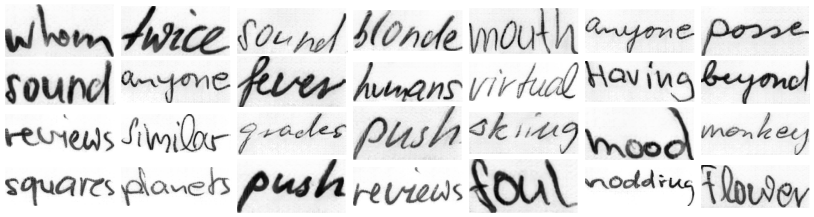


Fig. 4. Word image generation. a) In-Vocabulary (IV) words and seen (S) styles; b) In-Vocabulary (IV) words and unseen (U) styles; c) Out-of-Vocabulary (OOV) words and seen (S) styles and d) Out-of-Vocabulary (OOV) words and unseen (U) styles.

Table 1. FID between generated images and real images of corresponding set.

	Real images	IV-S	IV-U	OOV-S	OOV-U
FID	90.43	120.07	124.30	125.87	130.68

IAM transcriptions has been used to test the performance when producing OOV words. To quantitatively measure the image quality, diversity and the ability to transfer style attributes of the proposed approach we will use the Fréchet Inception Distance (FID) [17, 4], measuring the distance between the Inception-v3 activation distributions for generated $\bar{\mathcal{X}}$ and real samples $\mathcal{X}$ for each writer w_i separately, and finally averaging them. Inception features, trained over ImageNet data, have not been designed to discern between different handwriting images. Although this measure might not be ideal to evaluate our specific case, it will still serve as an indication of the similarity between generated and real images.

4.1 Generating Handwritten Word Images

We present in Fig. 4 an illustrative selection of generated handwritten words. We appreciate the realistic and diverse aspect of the produced images. Qualitatively, we observe that the proposed approach is able to yield satisfactory results even when dealing with both words and calligraphic styles never seen during training. But, when analyzing the different experimental settings in Table 1, we appreciate that the FID measure slightly degrades when either we input an OOV word or a style never seen during training. Nevertheless, the reached FID measures in all four settings satisfactorily compare with the baseline achieved by real data.

In order to show the ability of the proposed method to produce a diverse set of generated images, we present in Fig. 5 a t-SNE [31] visualization of different instances produced with a fixed textual content while varying the calligraphic style inputs. Different clusters corresponding to particular slants, stroke widths, character roundnesses, ligatures and cursive writings are observed.

To further demonstrate the ability of the proposed approach to coalesce content and style information into the generated handwritten word images, we compare in Fig. 6 our produced results with the outcomes of the state-of-the-art approach FUNIT [29]. Being an image-to-image translation method, FUNIT starts

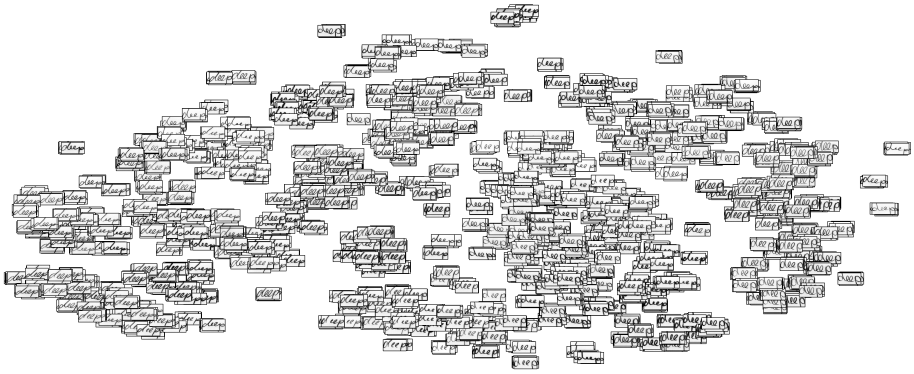


Fig. 5. t-SNE embedding visualization of 2.500 generated instances of the word `"deep"`.

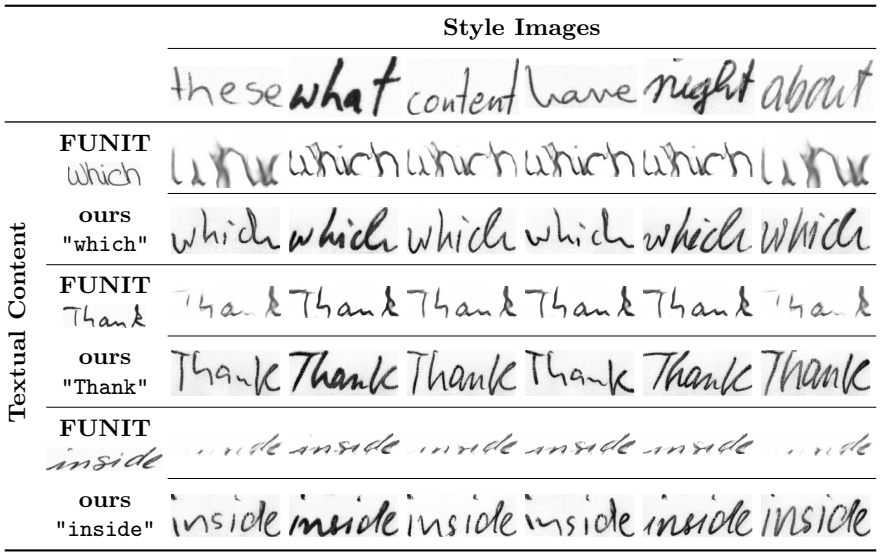


Fig. 6. Comparison of handwritten word generation with FUNIT [29].

with a content image and then injects the style attributes derived from a second sample image. Although FUNIT performs well for natural scene images, it is clear that such kind of approaches do not apply well for the specific case of handwritten words. Starting with a content image instead of a text string confines the generative process to the shapes of the initial drawing. When infusing the style features, the FUNIT method is only able to deform the stroke textures, often resulting in extremely distorted words. Conversely, our proposed generative process is able to produce realistic and diverse word samples given a content text string and a calligraphic style example. We observe how for the different produced versions of the same word, the proposed approach is able to change

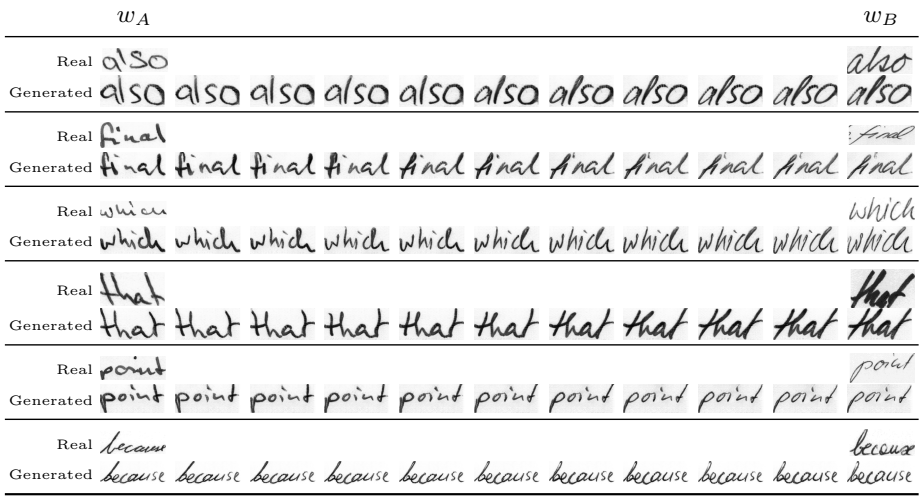


Fig. 7. Latent space interpolation between two calligraphic styles for different words while keeping contents fixed.

style attributes as stroke width or slant, to produce both cursive words, where all characters are connected through ligatures as well as disconnected writings, and even render the same characters differently, *e.g.* note the characters `n` or `s` in `"Thank"` or `"inside"` respectively.

4.2 Latent Space Interpolations

The generator network G learns to map feature points F in the latent space to synthetic handwritten word images. Such latent space presents a structure worth exploring. We first interpolate in Fig. 7 between two different points F_s^A and F_s^B corresponding to two different calligraphic styles w_A and w_B while keeping the textual contents t fixed. We observe how the generated images smoothly adjust from one style to another. Again note how individual characters evolve from one typography to another, *e.g.* the `l` from `"also"`, or the `f` from `"final"`.

Contrary to the continuous nature of the style latent space, the original content space is discrete in nature. Instead of computing point-wise interpolations, we present in Fig. 8 the obtained word images for different styles when following a "word ladder" puzzle game, *i.e.* going from one word to another, one character difference at a time. Here we observe how different contents influence stylistic aspects. Usually `s` and `i` are disconnected when rendering the word `"sired"` but often appear with a ligature when jumping to the word `"fired"`.

4.3 Impact of the Learning Objectives

Along this paper, we have proposed to guide the generation process by three complementary goals. The discriminator loss $\mathcal{L}_d$ controlling the genuine appearance of the generated images $\bar{x}$. The writer classification loss $\mathcal{L}_w$ forcing $\bar{x}$ to

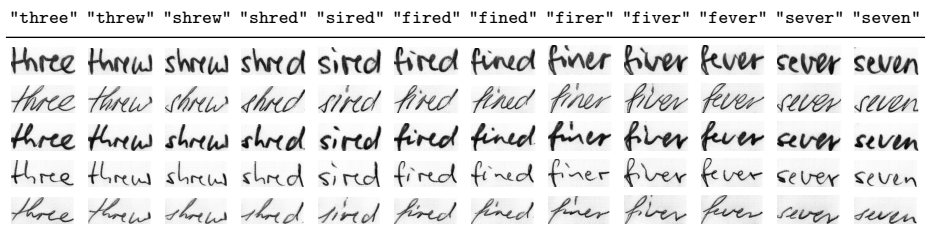


Fig. 8. Word ladder. From `"three"` to `"seven"` changing one character at a time, generated for five different calligraphic styles.

Table 2. Effect of each different learning objectives when generating the content $t =$ `"vision"` for different styles.

$\mathcal{L}_d$	$\mathcal{L}_w$	$\mathcal{L}_r$	FID	Style Images			
				these	what	have	about
✓	-	-	364.10				
✓	✓	-	207.47				
✓	-	✓	138.80				
✓	✓	✓	**130.68**				

mimic the calligraphic style of input images X_i. The recognition loss $\mathcal{L}_r$ guaranteeing that $\bar{x}$ is readable and conveys the exact text information t. We analyze in Table 2 the effect of each learning objective.

The sole use of the $\mathcal{L}_d$ leads to constantly generating an image that is able to fool the discriminator. Although the generated image looks like handwritten strokes, the content and style inputs are ignored. When combining the discriminator with the auxiliary task of writer classification $\mathcal{L}_w$, the produced results are more encouraging, but the input text is still ignored, always tending to generate the word `"the"`, since it is the most common word seen during training. When combining the discriminator with the word recognizer loss $\mathcal{L}_r$, the desired word is rendered. However, as in [1], we suffer from the mode collapse problem, always producing unvarying word instances. When combining the three learning objectives we appreciate that we are able to correctly render the appropriate textual content while mimicking the input styles, producing diverse results. We appreciate that the FID measure also decreases for each successive combination.

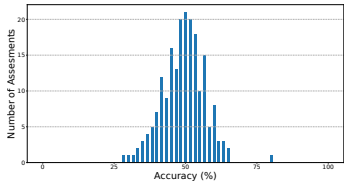
4.4 Human Evaluation

Finally, we also tested whether the generated images were actually indistinguishable from real ones by human judgments. We have conducted a human evaluation study as follows: we have asked 200 human examiners to assess whether a set

Table 3. Human evaluation plausibility experiment.

Actual	Predicted		
	Real	Fake	
Genuine	27.01	22.99	R: 54.1
Generated	27.69	22.31	FPR: 55.4
	P: 49.4	FOR: 50.8	ACC: 49.3

a) Confusion matrix (%)



b) Accuracy distribution

of images were written by a human or artificially generated. Appraisers were presented a total of sixty images, one at a time, and they had to choose if each of them was real of fake. We chose thirty real words from the IAM test partition from ten different writers. We then generated thirty artificial samples by using OOV textual contents and by randomly taking the previous writers as the sources for the calligraphic styles. Such sets were not curated, so the only filter was that the generated samples had to be correctly transcribed by the word recognizer network R. In total we collected $12,000$ responses. In Table 3 we present the confusion matrix of the human assessments, with Accuracy (ACC), Precision (P), Recall (R), False Positive Rate (FPR) and False Omission Rate (FOR) values. The study revealed that our generative model was clearly perceived as plausible, since a great portion of the generated samples were deemed genuine. Only a 49.3% of the images were properly identified, which shows a similar performance than a random binary classifier. Accuracies over different examiners were normally distributed. We also observe that the synthetically generated word images were judged more often as being real than correctly identified as fraudulent, with a final FPR of 55.4%.

5 Conclusion

We have presented a novel image generation architecture that produces realistic and varied artificially rendered samples of handwritten words. Our pipeline can yield credible word images by conditioning the generative process with both calligraphic style features and textual content. Furthermore, by jointly guiding our generator with three different cues: a discriminator, a style classifier and a content recognizer, our model is able to render any input word, not depending on any predefined vocabulary, while incorporating calligraphic styles in a few-shot setup. Experimental results demonstrate that the proposed method yields images with such a great realistic quality that are indistinguishable by humans.

Acknowledgements

This work was supported by EU H2020 SME Instrument project 849628, the Spanish projects TIN2017-89779-P and RTI2018-095645-B-C21, and grants 2016-DI-087, FPU15/06264 and RYC-2014-16831. Titan GPU was donated by NVIDIA.

References

1. Alonso, E., Moysset, B., Messina, R.: Adversarial generation of handwritten text images conditioned on sequences. In: Proceedings of the International Conference on Document Analysis and Recognition (2019)
2. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content gan for few-shot font style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7564–7573 (2018)
3. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc. (2009)
4. Borji, A.: Pros and cons of GAN evaluation measures. Computer Vision and Image Understanding **179**, 41–65 (2019)
5. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: Proceedings of the International Conference on Learning Representations (2019)
6. Chang, B., Zhang, Q., Pan, S., Meng, L.: Generating handwritten Chinese characters using CycleGAN. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (2018)
7. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
8. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: Proceedings of the NeurIPS Workshop on Deep Learning (2014)
9. Dong, H.W., Hsiao, W.Y., Yang, L.C., Yang, Y.H.: MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
10. Fogel, S., Averbuch-Elor, H., Cohen, S., Mazor, S., Litman, R.: Scrabblegan: Semi-supervised varying length handwritten text generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4324–4333 (2020)
11. Ganin, Y., Kulkarni, T., Babuschkin, I., Eslami, S., Vinyals, O.: Synthesizing programs for images using reinforced adversarial learning. In: Proceedings of the International Conference on Machine Learning (2018)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the Neural Information Processing Systems Conference (2014)
13. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
14. Gregor, K., Danihelka, I., Graves, A., Rezende, D.J., Wierstra, D.: DRAW: A recurrent neural network for image generation. In: Proceedings of the International Conference on Machine Learning (2015)
15. Ha, D., Eck, D.: A neural representation of sketch drawings. In: Proceedings of the International Conference on Learning Representations (2018)
16. Haines, T.S., Mac Aodha, O., Brostow, G.J.: My text in your handwriting. ACM Transactions on Graphics **35**(3), 1–18 (2016)
17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: Proceedings of the Neural Information Processing Systems Conference (2017)

18. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
19. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (2018)
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the International Conference on Machine Learning (2015)
21. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
22. Jiang, H., Yang, G., Huang, K., Zhang, R.: W-net: one-shot arbitrary-style Chinese character generation with deep neural networks. In: Proceedings of the International Conference on Neural Information Processing (2018)
23. Kang, L., Rusiñol, M., Fornés, A., Riba, P., Villegas, M.: Unsupervised adaptation for synthetic-to-real handwritten word recognition. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision. pp. 3491–3500 (2020)
24. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
25. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: Proceedings of the International Conference on Learning Representations (2014)
26. Konidaris, T., Gatos, B., Ntzios, K., Pratikakis, I., Theodoridis, S., Perantonis, S.J.: Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. International Journal of Document Analysis and Recognition **9**(2-4), 167–177 (2007)
27. Krishnan, P., Jawahar, C.: Generating synthetic data for text recognition. arXiv preprint arXiv:1608.04224 (2016)
28. Lin, Z., Wan, L.: Style-preserving English handwriting synthesis. Pattern Recognition **40**(7), 2097–2109 (2007)
29. Liu, M.Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
30. Lyu, P., Bai, X., Yao, C., Zhu, Z., Huang, T., Liu, W.: Auto-encoder guided GAN for Chinese calligraphy synthesis. In: Proceedings of the International Conference on Document Analysis and Recognition (2017)
31. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008)
32. Marti, U.V., Bunke, H.: The IAM-database: an English sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition **5**(1), 39–46 (2002)
33. Mayr, M., Stumpf, M., Nikolaou, A., Seuret, M., Maier, A., Christlein, V.: Spatio-temporal handwriting imitation. arXiv preprint arXiv:2003.10593 (2020)
34. Michael, J., Labahn, R., Grüning, T., Zöllner, J.: Evaluating sequence-to-sequence models for handwritten text recognition. arXiv preprint arXiv:1903.07377 (2019)
35. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
36. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: Proceedings of the International Conference on Machine Learning (2017)

37. Pondenkandath, V., Alberti, M., Diatta, M., Ingold, R., Liwicki, M.: Historical document synthesis with generative adversarial networks. In: Proceedings of the International Conference on Document Analysis and Recognition (2019)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations (2015)
39. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. In: Proceedings of the International Conference on Learning Representations (2017)
40. Thomas, A.O., Rusu, A., Govindaraju, V.: Synthetic handwritten CAPTCHAs. Pattern Recognition **42**(12), 3365–3373 (2009)
41. Tian, Y.: zi2zi: Master chinese calligraphy with conditional adversarial networks (2017), https://github.com/kaonashi-tyc/zi2zi
42. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: MoCoGAN: Decomposing motion and content for video generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
43. Wang, J., Wu, C., Xu, Y.Q., Shum, H.Y.: Combining shape and physical models for online cursive handwriting synthesis. International Journal of Document Analysis and Recognition **7**(4), 219–227 (2005)
44. Wu, S.J., Yang, C.Y., Hsu, J.Y.j.: Calligan: Style and structure-aware chinese calligraphy character generator. arXiv preprint arXiv:2005.12500 (2020)
45. Yu, L., Zhang, W., Wang, J., Yu, Y.: SeqGAN: Sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI Conference on Artificial Intelligence (2017)
46. Zhan, F., Xue, C., Lu, S.: Ga-dan: Geometry-aware domain adaptation network for scene text detection and recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9105–9115 (2019)
47. Zhan, F., Zhu, H., Lu, S.: Spatial fusion gan for image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3653–3662 (2019)
48. Zheng, N., Jiang, Y., Huang, D.: Strokenet: A neural painting environment. In: Proceedings of the International Conference on Learning Representations (2019)