Improving Handwritten OCR with Augmented Text Line Images Synthesized from Online Handwriting Samples by Style-Conditioned GAN

Mingyang Guan
Beijing Institute of Technology
Beijing, China, 100081
Email: Guanmingyangtf@126.com

Haisong Ding
University of Science and Technology of China
Hefei, China, 230026
Email: dinghs11@mail.ustc.edu.cn

Kai Chen and Qiang Huo
Microsoft Research Asia
Beijing, China, 100080
Email: {kaic, qianghuo}@microsoft.com

***Abstract*—By leveraging large amounts of training data and deep learning technologies, performances of modern handwritten optical character recognition (OCR) systems have been greatly improved. However, collecting and labeling massive handwriting images are both time-consuming and expensive. In this paper, we propose to augment handwritten OCR training with online handwriting samples. To achieve this goal, we propose a style-conditioned generative adversarial network (SC-GAN) with a novel training data pair generation strategy. Then this network is used to transfer the styles of real handwriting images to skeleton images extracted from online handwriting samples to generate photo-realistic text line images. Experimental results on a large scale handwritten OCR task show that the recognition accuracy of our handwritten OCR system is improved by using the augmented synthetic training data.**

I. Introduction

In recent years, the performances of handwritten optical character recognition (OCR) systems have been greatly improved [1], [2] by leveraging deep learning technologies including (a) recurrent neural networks (RNNs), especially deep bidirectional long short-term memory (DBLSTM) (e.g., [3], [4], [5], [6]) and multi-dimensional LSTM (MDLSTM) (e.g., [7], [8], [9]), and (b) convolutional neural networks (CNNs) (e.g., [10], [11]) and their combinations (e.g., [12], [13], [14], [15], [16]).

Same with other deep learning driven applications, large amounts of handwriting images are required to achieve excellent recognition accuracies for a handwritten OCR system. A representative training data set should cover various writing styles, colors, backgrounds, lighting conditions, vocabularies, etc. However, collecting and labelling handwriting images to build such a data set are both time-consuming and expensive. And privacy concerns should also be taken into account. As a result, building handwritten OCR systems with synthetic data could be a potential solution (e.g., [17]).

Synthetic text line images have been successfully applied to printed OCR tasks in natural scene image scenarios (e.g., [18], [19], [20], [21]). These images are generated by rendering text transcriptions with given typeset fonts, followed by colorization, distortion and natural scene image blending. Since natural handwriting styles are quite different from typeset fonts, these technologies cannot be used to build handwritten OCR systems.

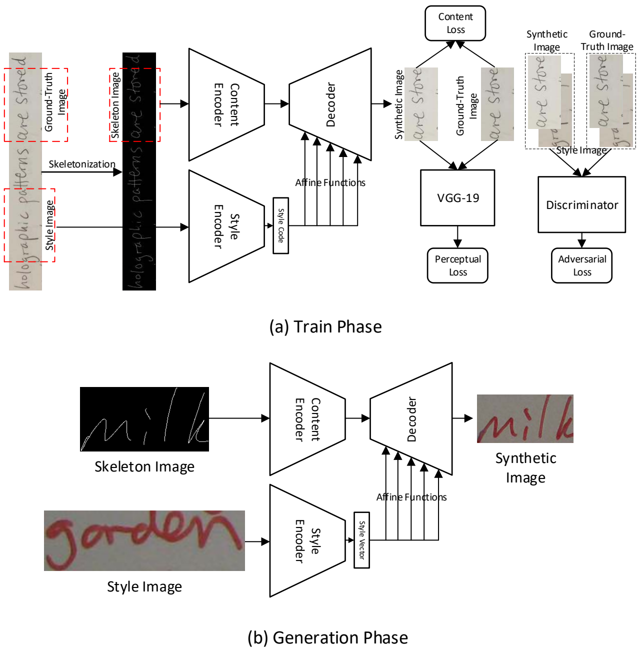
In the past few years, generative adversarial network (GAN) [22] based approaches (e.g., [23], [24], [25]) have achieved excellent performance in image synthesis area. These approaches are also applied to synthesize handwriting images. For example, [26] leveraged a text-to-image framework and tried to convert word transcriptions to text line images directly. However, the synthetic handwriting images they generated are not photo-realistic, which, as a result, can only slightly improve the recognition accuracy of their handwritten OCR system. The text-to-image framework is also adopted in [27]. Given a word transcription and a real offline image of a writer, it is able to generate realistic text lines that match the writing style of the writer.

Instead of generating handwriting samples from text, [17] tried to render online handwriting samples into offline images to augment handwritten OCR system training. The online handwriting samples can be real data or synthetic data using a sequence generation method [28]. Then these online handwriting samples are rendered to images using a similar pipeline as in [18]. Though these rendered images can be used to train handwritten OCR models, they can be easily distinguished from real text line images. Moreover, these images lack diversity in terms of styles (e.g., stroke widths, stroke and background textures, etc.).

To generate photo-realistic handwriting text lines from online handwriting samples with diverse styles, we propose a style-conditioned handwriting text synthesis system. This system is based on an image-to-image (I2I) translation framework [23] where online samples are first processed as binary skeleton images and then converted to text line images using adversarial training. To generate text line images with controllable and diverse styles, we encode the style (e.g., color, background texture, stroke width, stroke texture, etc.) of a real handwriting image as a style code and integrate it into the generator using an adaptive instance normalization (AdaIN) ([25], [29]) method. It is noted that handwriting trajectory styles are controlled by binary skeleton images.

This system only requires real handwriting image data for

978-1-7281-9966-5/20/$31.00 ©2020 IEEE
DOI 10.1109/ICFHR2020.2020.00037



(a) Train Phase

(b) Generation Phase

Fig. 1. Overview of our proposed handwriting text image synthesis system.

training and can make use of large amounts of online handwriting samples for generation. Given skeleton images extracted from online samples and real handwriting images encoded as style codes, our system can generate photo-realistic images that match the content of skeletons and the styles of real handwriting images. Experimental results on IAM handwriting English sentence dataset [30] and a large scale handwritten OCR task show that by combining real handwriting text lines with synthetic ones, we can improve the recognition accuracy of handwritten OCR systems.

The rest of the paper is organized as follows. In Section II, we present our handwriting text line generation system. In Section III, we evaluate the effectiveness of our system. Finally, we conclude this study in Section IV.

II. Our Approach

As shown in Fig. 1, our style-conditioned generator takes a skeleton image $\boldsymbol{I}_{\text{ske}}$ and a style image $\boldsymbol{I}_{\text{sty}}$ as inputs to generate a synthetic image $\boldsymbol{Y}_{syn}$, which is denoted as follows:

$$\boldsymbol{Y}_{syn} = G(\boldsymbol{I}_{\text{ske}}, \boldsymbol{I}_{\text{sty}}) \tag{1}$$

This synthetic image will contain the same content as the skeleton and follow the same style as the style image.

In order to train such a system, we need a training set containing matched skeleton, style and ground truth images. However, this kind of training set is difficult to collect since online handwriting data only contains ink trajectory sequences while offline data is only in image format, and they are collected with different devices. In this paper, we propose to use offline handwriting text line images to construct such a training set. As is shown in Fig. 1 (a), given an offline handwriting image, we first extract online ink trajectory sequences using a skeleton extraction tool in [31]. Then two non-overlapping image patches are randomly cropped. One serves as style image, and the other as ground truth image. The ink trajectory sequences corresponding to the ground truth image are rendered as a binary image which will play the role of skeleton. During training, content, perceptual and adversarial loss functions will be used to measure the difference between the synthetic image and the ground truth.

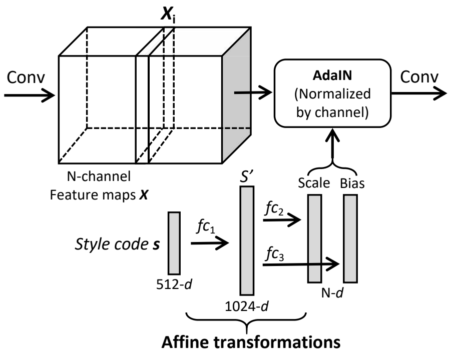


Fig. 2. Details of style integration structure.

In generation phase, given a skeleton image extracted from a rendered online handwriting sample, we will use an image sampled from real offline handwriting data set as a style image. Then our generator will produce a synthetic image that matches the content of skeleton image and the style of the style image as shown in Fig. 1 (b).

In following subsections, we will introduce the model architecture of our style-conditioned generative adversarial networks (SC-GAN) and training loss functions in detail.

A. Model Architecture

Our model employs an "encoder-decoder" structure as shown in Fig. 1. Given a binary skeleton image of height H and width W denoted as $\boldsymbol{I}_{\text{ske}}$, it is first processed by a content encoder. The network structure of the content encoder is borrowed from part of VGG-19 [32], which contains layers from "conv1-1" to "conv4-1". As a result, the input skeleton image will be converted to 512-channel feature maps, which are $\frac{H}{8}$-pixel high and $\frac{W}{8}$-pixel wide. Then, the output of the content encoder is processed by a decoder to generate a synthetic handwriting image $\boldsymbol{Y}_{\text{syn}}$. The decoder is symmetrical to its encoder counterpart, which has multiple 3×3 convolutional layers and up-sampling operations. Up-sampling is done by bi-linear interpolation with a scale factor of 2. The last convolutional layer will produce a synthetic image of height H and width W.

To integrate style information into the generator, given a real offline handwriting image as style image $\boldsymbol{I}_{\text{sty}}$, a style code is extracted using a style encoder. The style encoder is of the same structure as the content encoder, and a 512-dimensional style code $\boldsymbol{s}$ is obtained by using global average pooling operation on style encoder's output. Then this style code is used in an AdaIN operation applied for the output feature maps in all but the last convolutional layers of the decoder part.

Denote $\boldsymbol{X}$ as the output of a convolutional layer in decoder which has N feature maps. Similar to styleGan [25], style code $\boldsymbol{s}$ is first processed by a fully-connected layer with 1024 outputs to get $\boldsymbol{s}'$. Then two fully-connected layers are used to calculate scale and bias vectors $\boldsymbol{v}_\sigma$ and $\boldsymbol{v}_\mu$ respectively that characterize the style information in $\boldsymbol{s}$:

$$\boldsymbol{s}' = fc_1(\boldsymbol{s}) \tag{2}$$

$$\boldsymbol{v}_\sigma = fc_2(\boldsymbol{s}') \tag{3}$$

$$\boldsymbol{v}_\mu = fc_3(\boldsymbol{s}') \tag{4}$$

where fc_1, fc_2 and fc_3 represent three fully connected layers. In other words, $\boldsymbol{v}_\sigma$ and $\boldsymbol{v}_\mu$ are obtained by applying affine transformations to style code, and each convolutional layer has its own corresponding learnable affine transformations. Finally $\boldsymbol{v}_\sigma$ and $\boldsymbol{v}_\mu$ both have N elements and are used to re-normalize $\boldsymbol{X}$ using AdaIN operation calculated as follows:

$$\text{AdaIN}(\boldsymbol{X}, \boldsymbol{v}_\sigma, \boldsymbol{v}_\mu) = \boldsymbol{v}_\sigma \frac{\boldsymbol{X} - \mu(\boldsymbol{X})}{\sigma(\boldsymbol{X})} + \boldsymbol{v}_\mu \tag{5}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ represent mean and standard deviation function respectively. We plot details of this style integration structure in Fig. 2. By re-normalizing convolutional layers' outputs in decoder part, we expect that the synthesis generator will generate synthetic images that match the style of the given style image.

To make synthetic images more realistic, a GAN discriminator is used to guide the learning of the synthesis generator. The inputs of discriminator are the concatenation of style and synthetic (or real ground truth) images along the channel dimension. The concatenated images are processed by 3 convolutional blocks with Leaky ReLU [33] activation functions, each block containing a 3×3 convolution with a stride of 1 and a 4×4 convolution with a stride of 2 (used for downscaling). The output channels of these blocks are 64, 128 and 256, respectively. Finally, the output is followed by two 4×4 convolution of stride 1 with 512 and 1 output channels, respectively. As a result, the discriminator will produce a feature map of height $\frac{H}{8}$ and width $\frac{W}{8}$. Following the practice of patchGAN [23], every value of the $\frac{H}{8} \cdot \frac{W}{8}$ output is followed by a sigmoid function, thus predicting whether the corresponding sub-patch of input is synthetic or real.

During training, the discriminator will be tuned to distinguish real and synthetic input, whereas the generator will be trained to confuse the discriminator. As a result, synthetic images tend to be more realistic.

B. Training Loss Functions

The overall loss function contains three parts, namely content loss, perceptual loss [34] and adversarial loss, which is denoted as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{content}} + \lambda_2 \mathcal{L}_{\text{perceptual}} + \lambda_3 \mathcal{L}_{\text{adv}} \tag{6}$$

where λ's are weighting factors. The content loss will force synthetic and ground truth images to be more similar in general, the perceptual loss will make them produce similar low-level and high-level features given a pre-trained network, and the adversarial loss makes synthetic images more realistic.

Content loss: The basic feature of our handwriting text image synthesis model is that, given an online skeleton image $\boldsymbol{I}_{\text{ske}}$, it will produce a synthetic image $\boldsymbol{Y}_{\text{syn}}$ that matches the handwriting text in $\boldsymbol{I}_{\text{ske}}$. Therefore, we use an L_1 loss function to measure the pixel-wise difference between $\boldsymbol{Y}_{\text{syn}}$ and the ground truth image $\boldsymbol{Y}_{\text{gt}}$. We call it "content loss", which is calculated as follows:

$$\mathcal{L}_{\text{content}} = \|\boldsymbol{Y}_{\text{gt}} - \boldsymbol{Y}_{\text{syn}}\|_1 \tag{7}$$

Perceptual loss: Besides measuring their difference directly at pixel-level, we also expect them to produce similar low-level and high-level features. According to [34], minimizing low-level and high-level feature differences between $\boldsymbol{Y}_{\text{syn}}$ and $\boldsymbol{Y}_{\text{gt}}$ can ease the over-smoothed problem of L_1-based loss. Therefore, we use a pre-trained VGG-19 [32] on ImageNet dataset to produce low and high level features. In particular, we take the output functions of VGG-19's "relu1-1", "relu2-1", "relu3-1", "relu4-1" and "relu5-1" layers, and denote them as $\phi_i(\cdot), i = \{1, 2, \cdots, 5\}$, respectively. We call it "perceptual loss" which is formulated as follows:

$$\mathcal{L}_{\text{perceptual}} = \sum_{i=1}^{5} \alpha_i \cdot \text{MSE}\left(\phi_i(\boldsymbol{Y}_{\text{gt}}), \phi_i(\boldsymbol{Y}_{\text{syn}})\right) \tag{8}$$

where MSE is the function of mean squared error and the values of α's are 1/32, 1/16, 1/8, 1/4 and 1, respectively.

Adversarial loss: As discussed in Section II-A, we follow patchGAN [23] to increase discriminator's ability by computing an adversarial loss for every pixel in discriminator's output. The adversarial loss function is formulated as follows:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}[\log\left(D(\boldsymbol{I}_{\text{sty}}, \boldsymbol{Y}_{\text{gt}})\right)] + \mathbb{E}[\log\left(1 - D(\boldsymbol{I}_{\text{sty}}, \boldsymbol{Y}_{\text{syn}})\right)] \tag{9}$$

where $D(\cdot)$ denotes the discriminator.

The parameters of synthesis generator's encoder part are borrowed from pre-trained VGG-19 and kept fixed. During training, the generator is trained by minimizing Eq. (6) and the discriminator is trained by maximizing Eq. (9).

III. Experiments

A. Experimental Setup

We first evaluate our approach on IAM dataset [35]. The dataset contains 1,539 pages of scanned text (657 writers contributed samples of their handwritings). We only use the training set to train our synthesis models. We directly crop every text image row by row from the original page image to obtain 6161 text lines, and normalize them to a fixed height of H. When constructing mini-batches, two non-overlap patches of size $H \times W$ is randomly selected as ground truth and style image respectively. For corresponding skeleton images, we first use a skeleton extraction tool [31] on ground truth image to extract strokes, and then render these strokes using

TABLE I
EXAMPLES OF SYNTHETIC IMAGES GENERATED FROM THE SAME SKELETON IMAGE WITH DIFFERENT STYLE IMAGES ON IAM TASK.

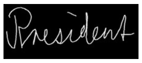
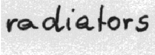
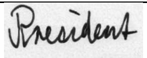
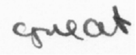
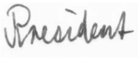
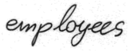
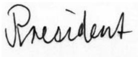
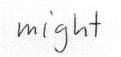
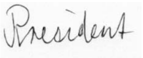
Skeleton	Style	Synthetic
		
		
		
		

TABLE II
FIDS BETWEEN REAL AND SYNTHETIC HANDWRITING IMAGES GENERATED BY DIFFERENT METHODS ON IAM TASK.

Method	FID-2048	FID-796	FID-192	FID-64
Rendering	88.30	1.11	33.08	7.12
Our approach	74.00	0.56	4.60	1.22

TABLE III
CER (IN %) AND WER (IN %) OF HANDWRITTEN OCR SYSTEMS TRAINED USING DIFFERENT TRAINING SETS ON IAM TASK.

Training set	Val		Test	
	CER	WER	CER	WER
Real	3.6	9.3	5.3	13.0
Synthetic (rendering)	46.9	61.6	49.7	65.8
Synthetic (our approach)	7.6	16.4	11.1	21.7
Real + Synthetic (our approach)	2.6	7.3	4.2	10.7

1-pixel wide white lines on a black background to get a binary image. In experiments, we use $H = 96$ and $W = 3H$.

During training, models are optimized using Adam [36] with $\beta_1 = 0.5$, $\beta_2 = 0.999$. We use $\lambda_1 = \lambda_2 = 1.0$ and $\lambda_3 = 0.2$ in Eq. (6). The initial learning rate (lr) is 0.0001 and we rescale lr by 0.1 every 20 epochs. Models are trained for 100 epochs and the final model is used to generate synthetic text line images.

During generation, skeleton images are extracted from IAM online handwriting dataset [30]. For line samples in IAM online dataset, we first render them to images using fixed-width strokes and then skeletons are extracted with the same tool as training. We get 12,195 skeleton images from this dataset in total. For style images, we use the same 6,161 training text line images. By randomly sampling skeleton and style images as inputs, the synthesis model will generate images that match the content of skeleton images and the style of style images.

B. *Quality Evaluation*

In Table I, we try to transfer the styles of different style images to the same skeleton image with our proposed approach. It shows that, the synthetic images are of similar stroke widths and foreground/background pixel distributions with styles while keeping its content same with skeleton.

To evaluate the quality of synthetic text line images quantitatively, 2 metrics are used. The first metric is Frchet Inception Distance (FID) [37] which measures the difference between real and synthetic images by calculating the distance between their distributions of high-level features. The second metric is the character error rate (CER) and word error rate (WER) of the handwritten OCR system trained with offline IAM training set (6,161 lines) and synthetic handwriting data (12,195 lines) on IAM validation (966 lines) and testing set (2,914 lines).

We calculate FID using features from the final average pooling layer (FID-2048), pre-auxiliary classifier layer (FID-796), second max pooling layer (FID-192) and first max pooling layer (FID-64) of a pre-trained Inception V3 network. As for WER and CER, we use the same VGG-DBLSTM based handwritten OCR system as in [13]. To show the effectiveness of our synthesis method, we compare it with a naive rendering approach where online handwriting trajectory sequences are simply drawn on an image with a fixed stroke width. Stroke and background colors are obtained using color clustering on given style images. For fair comparison, the same style-skeleton image pairs are used in experiments.

Table II lists the FIDs between real and synthetic images. Table III summarizes the recognition performance comparison of handwritten OCR systems trained with different datasets. When using synthetic data only, the simple rendering-based approach achieves far worse results. This is because the distributions of rendered images are quite different from real data as shown in Table II. With the improved similarity in terms of FIDs between synthetic and real data by our proposed approach, the performance of the handwritten OCR system trained on synthetic data can be improved greatly. However, it is still much worse than that trained with real data. One possible reason is that the writing styles of IAM offline data in test set are quite different from IAM online data due to the differences in collection method (writing on papers *vs.* writing on screens). Another possible reason is that the skeleton images extracted from real data are noisy, causing mismatches between training and generation.

Compared with the handwritten OCR system trained with real data only, combining our synthetic data with real data as a new training set achieves better recognition results. It achieves a relative CER reduction (CERR) of 27.8% and a relative WER reduction (WERR) of 21.5% on validation set. The CERR and WERR on testing set are 20.8% and 17.7%, respectively. Ahough adding synthetic data can improve the performance of handwritten OCR systems, real offline data is still very important in order to achieve excellent performances.

C. *Effects of Different Loss Functions*

To show the effects of 3 loss functions, we perform an ablation study by trying different loss combinations. Specifically, we try using $\mathcal{L}_{\text{content}}$ alone, and combining $\mathcal{L}_{\text{content}}$ with $\mathcal{L}_{\text{perceptual}}$. Then we generate synthetic images with the same style-skeleton pair setting using these 2 newly-trained models.

Visualizations of some synthetic images are listed in Table IV. We compare their FIDs in Table V. When only $\mathcal{L}_{\text{content}}$

TABLE IV
VISUALIZATION OF IMAGES GENERATED FROM SYNTHESIS MODELS WITH DIFFERENT LOSS FUNCTIONS ON IAM TASK (TO BETTER SHOW THE DIFFERENCE, WE HIGHLIGHT LETTER "A" IN "SAY" AND THE FIRST LETTER "E" IN "BELIEVES").

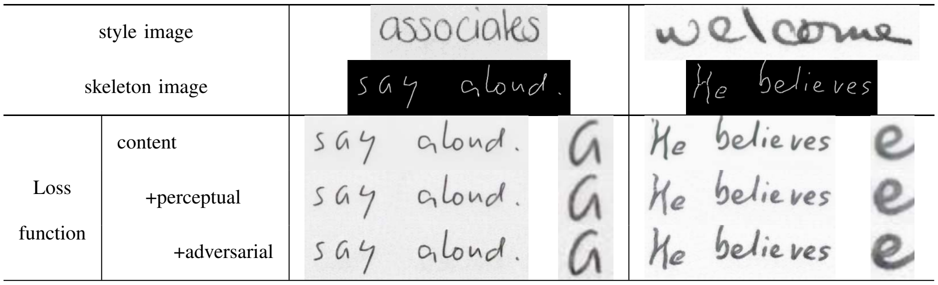

style image
skeleton image
Loss function
content
+perceptual
+adversarial
associates
welcome
say aloud.
He believes

TABLE V
FIDS BETWEEN REAL DATA AND SYNTHETIC IMAGES GENERATED FROM SYNTHESIS MODELS TRAINED WITH DIFFERENT LOSS FUNCTIONS ON IAM TASK.

Loss function	FID-2048	FID-796	FID-192	FID-64
content	87.16	0.70	16.37	3.68
+perceptual	71.57	0.57	6.59	2.02
+adversarial	74.00	0.56	4.60	1.22

TABLE VI
CER (IN %) AND WER (IN %) OF HANDWRITTEN OCR SYSTEMS TRAINED USING SYNTHETIC IMAGES GENERATED FROM SYNTHESIS MODELS WITH DIFFERENT LOSS FUNCTIONS ON IAM TASK.

Loss function	Val		Test	
	CER	WER	CER	WER
content	16.7	30.3	20.1	36.9
+perceptual	9.6	17.8	12.2	22.0
+adversarial	7.6	16.4	11.1	21.7

is used, the backgrounds of synthetic images are smooth and clean, which leads to poor FID scores indicating that there is a high mismatch between synthetic and real images. It is observed that $\mathcal{L}_{\text{perceptual}}$ and $\mathcal{L}_{\text{adv}}$ can help mitigate this problem to generate more realistic images that better match the styles in the corresponding style images.

We also compare the WERs/CERs of trained handwritten OCR systems. Experimental results are summarized in Table VI. The results show that using $\mathcal{L}_{\text{content}}$ alone achieves the worst recognition accuracy, which is consistent with the visualization results and FID scores.

D. Results on Large Scale Handwritten OCR Task

Next, we verify the effectiveness of our proposed approach on a large scale handwritten OCR task. We first build a synthesis model with 145k handwriting text line images extracted from whiteboard and handwritten notes. The mini-batch formation and hyper-parameters in training are the same as in IAM task except that W is set as $2H$. With this model, we render 506.8k online handwriting lines sampled from an in-house online handwriting dataset to text line images. Table VII gives some examples of synthetic images, whose styles look quite similar with the corresponding style images. It is hard to distinguish synthetic images from real images.

TABLE VII
EXAMPLES OF SYNTHETIC IMAGES GENERATED FROM VARIOUS SKELETON IMAGES WITH DIFFERENT STYLE IMAGES ON LARGE SCALE HANDWRITTEN OCR TASK.

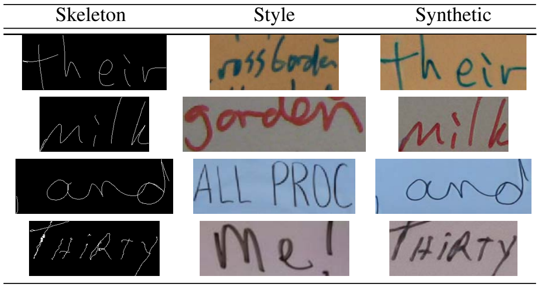

Skeleton
Style
Synthetic
their
rossGarden
their
milk
garden
milk
and
ALL PROC
and
THIRTY
Me!
THIRTY

TABLE VIII
CER (IN %) AND WER (IN %) OF HANDWRITTEN OCR SYSTEMS TRAINED USING DIFFERENT TRAINING SETS ON LARGE SCALE HANDWRITTEN OCR TASK.

Training set	E2E		IAM	
	CER	WER	CER	WER
Real	4.0	12.9	3.7	8.7
Real + Synthetic (our approach)	3.7	12.4	3.1	7.7

Then we combine the synthetic dataset with a large scale real dataset to build handwritten OCR systems. The real dataset contains about 248K handwriting text line images extracted from whiteboard and handwritten note images as training set. We also use a validation set which contains 28k text line images to guide model training. During evaluation, 2 test sets are used, one is for end-to-end testing called "E2E" with 3,953 text lines, the other is from IAM testing set called "IAM" with 1,861 text lines. As for the handwritten OCR system, a compact DarkNet-DBLSTM character model as in [38] is used. Results are summarized in Table VIII. We observe that adding synthetic data improves the recognition result on both testing sets, especially on the "IAM" testing set. This is because the writing styles of online handwriting samples we use are similar with that of the "IAM" testing set. These results show that with the augmented synthetic data, the performance of the handwritten OCR system can be improved.

IV. Conclusion

In this paper, we propose a style-conditioned GAN (SC-GAN) that can transfer styles of real handwriting images to skeleton images extracted from online handwriting samples. With a novel training set generation strategy, we can successfully build such a model and synthesize photo-realistic handwriting images. Experimental results show that when combined with real data, these synthetic images can improve the performance of handwritten OCR systems significantly. The proposed approach provides a potential solution to constructing high performance handwritten OCR systems without collecting and labeling massive real handwritten images. As future work, we will investigate better skeletonization method and improve synthetic image quality by better training strategies and more advanced model architectures.

Declaration

This work was done when Mingyang Guan and Haisong Ding worked as interns in Speech Group, Microsoft Research Asia, Beijing, China.

References

[1] J. A. Sanchez, V. Romero, A. H. Toselli, M. Villegas, and E. Vidal, "ICDAR2017 competition on handwritten text recognition on the READ dataset," in *Proceedings of ICDAR*, 2017, pp. 1383–1388.

[2] T. Strau, G. Leifert, R. Labahn, T. Hodel, and G. Mhlberger, "ICFHR2018 competition on automated text recognition on a READ dataset," in *Proceedings of ICFHR*, 2018, pp. 477–482.

[3] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[4] T. Bluche, H. Ney, J. Louradour, and C. Kermorvant, "Framewise and CTC training of neural networks for handwriting recognition," in *Proceedings of ICDAR*, 2015, pp. 81–85.

[5] K. Chen, Z.-J. Yan, and Q. Huo, "A context-sensitive-chunk BPTT approach to training deep LSTM/BLSTM recurrent neural networks for offline handwriting recognition," in *Proceedings of ICDAR*, 2015, pp. 411–415.

[6] Q. Liu, L.-J. Wang, and Q. Huo, "A study on effects of implicit and explicit language model information for DBLSTM-CTC based handwriting recognition," in *Proceedings of ICDAR*, 2015, pp. 461–465.

[7] B. Moysset, T. Bluche, K. Maxime, M. F. Benzeghiba, R. Messina, J. Louradour, and C. Kermorvant, "The A2iA multilingual text recognition system at the second Maurdor evaluation," in *Proceedings of ICFHR*, 2014, pp. 297–302.

[8] P. Voigtlaender, P. Doetsch, and H. Ney, "Handwriting recognition with large multidimensional long short-term memory recurrent neural networks," in *Proceedings of ICFHR*, 2016, pp. 228–233.

[9] B. Moysset and R. Messina, "Are 2D-LSTM really dead for offline text recognition?" *International Journal on Document Analysis and Recognition*, vol. 22, no. 1, pp. 193–208, 2019.

[10] D. Suryani, P. Doetsch, and H. Ney, "On the benefits of convolutional neural network combinations in offline handwriting recognition," in *Proceedings of ICFHR*, 2016, pp. 193–198.

[11] S. Wang, L. Chen, L. Xu, W. Fan, J. Sun, and S. Naoi, "Deep knowledge training and heterogeneous CNN for handwritten Chinese text recognition," in *Proceedings of ICFHR*, 2016, pp. 84–89.

[12] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[13] H. Ding, K. Chen, Y. Yuan, M. Cai, L. Sun, S. Liang, and Q. Huo, "A compact CNN-DBLSTM based character model for offline handwriting recognition with Tucker decomposition," in *Proceedings of ICDAR*, 2017, pp. 507–512.

[14] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in *Proceedings of ICDAR*, 2017, pp. 67–72.

[15] F. Cong, W. Hu, Q. Huo, and L. Guo, "A comparative study of attention-based encoder-decoder approaches to natural scene text recognition," in *Proceedings of ICDAR*, 2019, pp. 916–921.

[16] Y. Huang, C. Luo, L. Jin, Q. Lin, and W. Zhou, "Attention after attention: Reading text in the wild with cross attention," in *Proceedings of ICDAR*, 2019, pp. 274–280.

[17] R. R. Ingle, Y. Fujii, T. Deselaers, J. Baccash, and A. C. Popat, "A scalable handwritten text recognition system," in *Proceedings of ICDAR*, 2019, pp. 17–24.

[18] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Workshop on Deep Learning, Proceedings of NIPS*, 2014.

[19] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of CVPR*, 2016, pp. 2315–2324.

[20] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[21] D. Etter, S. Rawls, C. Carpenter, and G. Sell, "A synthetic recipe for OCR," in *Proceedings of ICDAR*, 2019, pp. 864–869.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of NIPS*, 2014, pp. 2672–2680.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of CVPR*, 2017, pp. 5967–5976.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of ICCV*, 2017, pp. 2242–2251.

[25] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of CVPR*, 2019, pp. 4401–4410.

[26] E. Alonso, B. Moysset, and R. O. Messina, "Adversarial generation of handwritten text images conditioned on sequences," in *Proceedings of ICDAR*, 2019, pp. 481–486.

[27] L. Kang, P. Riba, Y. Wang, M. Rusiol, A. Forns, and M. Villegas, "GANwriting: Content-conditioned generation of styled handwritten word images," *arXiv:2003.02567*, 2020.

[28] A. Graves, "Generating sequences with recurrent neural networks," *arXiv:1308.0850*, 2013.

[29] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of ICCV*, 2017, pp. 1510–1519.

[30] M. Liwicki and H. Bunke, "IAM-OnDB an on-line english sentence database acquired from handwritten text on a whiteboard," in *Proceedings of ICDAR*, 2005, pp. 956–961.

[31] X. Zhang, M. Wang, L. Wang, Q. Huo, and H. Li, "Building handwriting recognizers by leveraging skeletons of both offline and online samples," in *Proceedings of ICDAR*, 2015.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of ICLR*, 2015.

[33] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Workshop on Deep Learning for Audio, Speech and Language Processing, Proceedings of ICML*, 2013.

[34] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of ECCV*, 2016, pp. 694–711.

[35] U. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, pp. 39–46, 2002.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proceedings of NIPS*, 2017, pp. 6626–6637.

[38] H. Ding, K. Chen, and Q. Huo, "Compressing CNN-DBLSTM models for OCR with teacher-student learning and Tucker decomposition," *Pattern Recognition*, vol. 96, no. 1, p. 106957, 2019.