Distilling Content from Style for Handwritten Word Recognition

Lei Kang[*†], Pau Riba[*], Marçal Rusiñol[*], Alicia Fornés[*], Mauricio Villegas[†]
[*]Computer Vision Center, Computer Science Dept., Universitat Autònoma de Barcelona, Spain
{lkang, priba, marcal, afornes}@cvc.uab.es
[†]omni:us, Berlin, Germany
{lei, mauricio}@omnius.com

***Abstract*—Despite the latest transcription accuracies reached using deep neural network architectures, handwritten text recognition still remains a challenging problem, mainly because of the large inter-writer style variability. Both augmenting the training set with artificial samples using synthetic fonts, and writer adaptation techniques have been proposed to yield more generic approaches aimed at dodging style unevenness. In this work, we take a step closer to learn style independent features from handwritten word images. We propose a novel method that is able to disentangle the content and style aspects of input images by jointly optimizing a generative process and a handwritten word recognizer. The generator is aimed at transferring writing style features from one sample to another in an image-to-image translation approach, thus leading to a learned content-centric features that shall be independent to writing style attributes. Our proposed recognition model is able then to leverage such writer-agnostic features to reach better recognition performances. We advance over prior training strategies and demonstrate with qualitative and quantitative evaluations the performance of both the generative process and the recognition efficiency in the IAM dataset.**

***Index Terms*—Handwritten word recognition, content and style disentanglement, image-to-image translation, handwriting generation, sequence-to-sequence neural networks.**

I. Introduction

Handwritten Text Recognition (HTR) is one of the most fundamental tasks in the document analysis community and has been studied for decades. However, even with the latest advances in deep learning, HTR is still challenging due to the unlimited variations of handwriting styles, both when dealing with multiple writers, but also among samples from the same person.

Sometimes, it is hard even for humans to interpret handwritten texts. Even if it is a cliché, most of us have struggled to decipher what our doctor has written in a prescription. And, on occasions, if we have been sloppy when taking notes, we even have to strive to understand our own handwriting. From that point of view, handwriting recognition can even be seen as a much challenging task than other problems related to vision, such as image classification or semantic segmentation, since us humans are more error-prone when reading handwritten text than when asked to identify and pinpoint familiar objects in pictures. Luckily, words do not appear out of context, so the recognition processes are often leaded by the use of dictionaries and language models that are able to guide the visual decoding of the word sequence.

Nonetheless, having *the* automatic text recognizer that is able to correctly recognize samples from whatever writer, independently of its writing style, is a goal eagerly pursued. In order to accomplish that, such recognizer shall be able to extract features encoding the essential information to decode the textual contents of the handwritten text images while completely disregarding the features that describe the visual appearance of the writing samples. Such dissociation between textual content and writing style shall hopefully yield a generic recognition system that is able to easily cope with the inter-writer variability that we are often confronted with. But how to learn content features that are completely disentangled from the writing style?

In the last years, with the advent of adversarial strategies, there has been several quantum leaps in the quality of generative and style transfer approaches. We are now able to artificially depict a given image as if it was painted by van Gogh, Munch, Cézanne or Picasso [1]. In order to generate such make-believe paintings, the current generative approaches are forced to disentangle the features that contribute to encode the contents of the images from the features that just describe the visual style. Consequently, such approaches shall be able to distill the calligraphic style characteristics from the actual character sequence that conceals the textual contents, when applied to the handwriting image scenario.

In this paper we will work with the hypothesis that we shall be able to learn better writer independent features for Handwritten Word Recognition (HWR) if we use a generative style and content transfer process as an auxiliary proxy task. Such generative transfer process has the objective to produce images that convey the same textual content from an input image but that imitate the writing style of another sample image. In order to succeed in this task, the generative network has to learn to disentangle content attributes as much as possible from the producing writing style. Therefore, such content features that have been uncoupled from visual writing style characteristics, shall serve to yield a handwritten word recognizer that is as much independent of the different writing styles as possible.

Specifically, in this paper we propose to use an adversarial generative process that is able to mix and match content and visual writing styles from pairs of word image samples. The generative process is guided by three different objectives, namely, producing realistic images, that convey a certain textual content while having correctly transferred the writing style characteristics from one writer to another. But producing realistic images is not the ultimate goal of our approach. The generation of artificial images is just used as an auxiliary proxy task aimed at learning content features that have been completely disentangled from writing styles to be later fed to a handwritten word recognizer. In this paper we implement such recognition scheme as a sequence-to-sequence network architecture guided by an attention mechanism. The whole system, generator and recognizer, is trained in an end-to-end fashion, and our experimental results using the IAM dataset prove that the yielded content features are able to outperform other state-of-the-art feature training strategies. The main novelty and contribution of the proposed approach is the combination of a generative process able to disentangle content from style together with a handwritten word recognizer, that, when optimized in an end-to-end manner, is able to learn style independent content features.

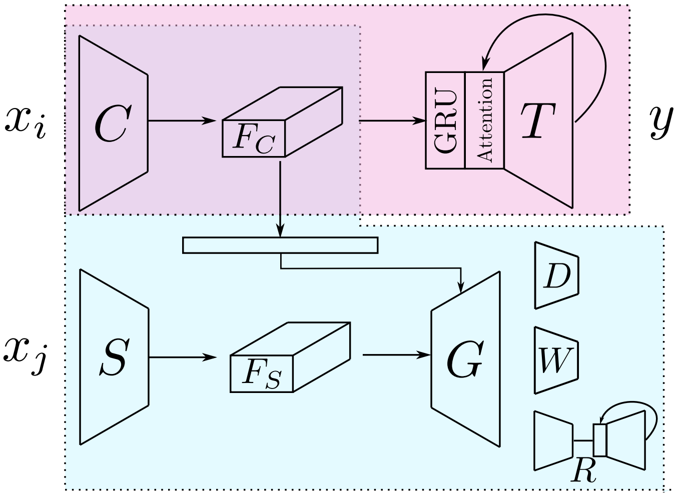


Fig. 1. Architecture of the proposed model with HWR module in red box and disentanglement module in blue box. Note that the content encoder C is used by both the HWR module and the disentanglement module.

II. Related Work

Being a sequential signal in nature, the recognition of handwritten text has usually been approached by statistical pattern recognition techniques that are able to cope with scenarios where the sample size is not fixed in advance. While the state of the art was driven by Hidden Markov Models (HMM) years ago [2]–[4], the rise of Deep Learning turned Bidirectional Long Short-Term Memory (BLSTM) networks [5]–[8] into the preferred standard. However, inspired by huge success of the recent automatic translation systems [9], [10], sequence-to-sequence approaches designed by stacking encoder and decoder networks powered with attention mechanisms have recently been proposed for the particular problem of recognizing handwritten text [11], [12]. Specifically, we will base our proposed recognition approach in such networks.

But in order to efficiently train such deep learning based approaches, a huge amount of labeled training data is required. Even when having tens of thousands of training samples, it is often still difficult to get good recognition performances on target data because of the different inter- and intra- writing style variations. To solve this problem and boost the performance, some approaches [13]–[15] proposed to alleviate the training data needs by using synthetically generated cursive data with electronic true-type fonts. Although it involves an important increase on computational costs during the training process, with such a strategy, one can have unlimited annotated data for free and train models that are less prone to overfit to a set of specific writing styles. Therefore, real target data can be reserved to a final fine-tuning step. Such approaches yield good performances but are somehow limited to the realistic aspect of the synthetically generated training samples. In order to make use of the data wisely and try to tune the recognizer precisely for the target data, writer adaptation approaches [16], [17] have been proposed to specifically adapt generic handwritten word recognizers to a specific handwriting style. The advantage of such approaches is that they are able to work in an unsupervised manner, without needing labeled data for the target writer domain. However, such fine-tuned adaptation to specific writers has to be computed *ad-hoc* per each writer we want to tackle, instead of having a writer-independent recognizer.

With the emergence of Generative Adversarial Networks (GANs) [18] and their use for image-to-image translation problems, we can find several recent works focused on separating content from style features. Most image-to-image translation approaches [1], [19], [20] are able to disentangle content information from style characteristics by using unpaired image data. They are then able to generate new artificial images that combine different content and style pairs. However, such approaches have been usually engineered to transfer artistic styles from images, *e.g.* painting styles, sketched aspect, etc. and have scarcely been used to text [21], [22]. In such approaches, the generated images were just used as a synthetic generator to gain training volume, mostly for scene text detection. Wang *et al.* [23] proposed a method for scene text recognition by separating font features from semantic features, which shares the same goal as us. However in their method, paired data is mandatory to train the model properly and the size of the input images need to be fixed. Thus, it cannot handle the variable length of handwritten data, nor the scarce data setting without paired information. To the best of our knowledge, this is the first attempt at jointly incorporating the concept of generative content and style disentanglement into a handwritten word recognition process in an end-to-end manner.

III. Handwritten Word Recognizer by Disentanglement

A. 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 $\mathcal{W} = \{w_i\}_{i=1}^N$. Given an image $x_i \in \mathcal{X}$ the proposed encoder-decoder Handwritten Word Recognizer (HWR) model has the ability to disentangle the textual content information by means of a style-agnostic encoder C. To this end, our HWR architecture is trained jointly with an auxiliary generative component able to disentangle the style and content features. The overall architecture is shown in Figure 1, which consists of two pipelines, on the one hand, a disentanglement module (blue box) and on the other hand, the HWR module (red box). Given a pair of handwritten word images (x_i, x_j) from different writers, the disentanglement module is aimed at generating handwritten word images having the textual content from x_i and sharing the calligraphic style attributes with x_j, while the HWR module will use the same content features in order to transcribe the incoming word x_i.

B. Disentanglement Network

Given a pair of handwritten word images (x_i, x_j), the proposed disentanglement module consists of two dedicated encoders. The style encoder S and the textual content encoder C modules are followed by a conditioned image generator G. Both, the content encoder C and the style encoder S are implemented with a VGG-19-BN network [24] with pre-trained weights from ImageNet. Despite being defined with the same architecture, their roles are completely different. On the one hand, C is devoted to the textual content information of x_i. The obtained features F_c are thus style-agnostic. On the other hand, S encodes the calligraphic style features F_s of the input word image x_j regardless of its textual content.

Given the encoded features F_c and F_s for the content and the style respectively, the generator G combines them to generate a new handwritten word image. It consists of two residual blocks [25] with AdaIN [26] normalization layers, followed by four convolutional modules with nearest neighbor up-sampling, with a final $\tanh$ activation layer.

The content features F_c are injected to the generator G by means of the AdaIN layers. In that sense, F_c is firstly squeezed into a one dimensional vector and processed via three fully connected layers, *viz.* $\bar{F_c}$. And this process from input image x_i to the one dimensional content vector $\bar{F_c}$ is denoted by $\bar{C}$. Thus, G combines the content feature $\bar{F_c}$ and the style feature F_s together to generate the output image $\bar{x}_{ij}$. AdaIN is formally defined as

$$\mathrm{AdaIN}\left(z, \alpha, \beta\right) = \alpha \left(\frac{z - \mu\left(z\right)}{\sigma\left(z\right)}\right) + \beta, \tag{1}$$

where μ and σ are the channel-wise mean and standard deviations. The parameters α and β are obtained by splitting $\bar{F_c}$ in four pairs, which are then used as the α and β of each four AdaIn layers in our generator architecture.

The whole disentanglement process is defined as

$$\bar{x}_{ij} = G(\bar{F_c}, F_s) = G(\bar{C}(x_i), S(x_j)), \tag{2}$$

where $x_i, x_j \in \mathcal{X}$. Thus, $\bar{x}_{ij}$ is expected to contain the same textual content of x_i and to share calligraphic style attributes with x_j.

C. Handwritten Word Recognition

In this work, an encoder-decoder architecture topped with an attention mechanism has been adopted as our handwritten word recognizer [11], [12], [27] as shown in Figure 1. Thus, our framework is divided into two components, the image encoder and the attention-based decoder.

Image encoder. Given a handwritten word image, the image encoder extracts high-level features that will further be transcribed as text strings in the decoder stage. The proposed encoder consists of two components, firstly, the Content Encoder C that is reused from the disentanglement module in order to obtain style-agnostic features; and secondly, a Recurrent Neural Network (RNN) which provides additional positional information to the final feature representation. In this work, the RNN is a multi-layered Bi-directional Gated Recurrent Unit (GRU).

Attention-based Decoder. Following the traditional architectures for sequence-to-sequence models, our decoder T is defined as a one-directional multi-layered GRU. The decoder predicts one character $\hat{y}_k$ at each time step k until meeting the end of sequence symbol $\langle \mathrm{end} \rangle$ or reaching the maximum number of steps K. Moreover, a location-based attention [28] has been adopted to align our input features with the expected output. Therefore, our attention module learns to process the features according to its positional order, which follows the nature of handwriting in latin script, *i.e.* from left to right.

Finally, the whole HWR framework is defined as a combination of these modules

$$\hat{y} = T(\mathrm{GRU}(F_c)) = T(\mathrm{GRU}(C(x_i))), \tag{3}$$

where $x_i \in \mathcal{X}$. Note that C only encodes textual content information as a result of the proposed disentanglement network. Thus, the obtained features shall ease task of the decoder in obtaining the proper textual transcriptions in multi-writer scenarios with large inter-writer visual style variability.

D. End-to-end Learning

The proposed system is jointly trained for the recognizer and the disentanglement module following an end-to-end fashion. Furthermore, four objective functions are proposed to guide our training process, namely, word recognition loss, discriminator loss, writer classification loss and content recognition loss.

On the one hand, our main HWR module is trained following the classic recognition loss function in an encoder-decoder framework, *i.e.* the Kullback-Leibler divergence loss. Thus, the loss function is formally defined as

$$\mathcal{L} = -\mathbb{E}_{x\sim\mathcal{X}} \left[\sum_{i=0}^{l} \sum_{j=0}^{|\mathcal{A}|} y_{i,j} \log \left(\frac{y_{i,j}}{\hat{y}_{i,j}} \right) \right], \tag{4}$$

where $\hat{y}$ is the predicted string given an image $x \in \mathcal{X}$; therefore, $\hat{y}_i$ corresponds to the i-th decoded character probability, $\hat{y}_{i,j}$ is the probability of j-th symbol in our alphabet $\mathcal{A}$ for $\hat{y}_i$, and $y_{i,j}$ is the real probability provided as a ground-truth. The empty symbol ε is ignored in the loss computation.

On the other hand, the disentanglement module is guided by three learning objectives which impose different constrains.

Discriminator Loss. Our discriminator D consists of six residual blocks with LeakyReLU activations and average pooling, followed by a binary classification layer, which tries to discriminate real images from generated ones, optimizing a *min-max* problem as proposed for adversarial strategies. Therefore, we make use of a binary cross entropy loss as an objective function. This module is in charge to guide the general visual appearance of the generated words $\bar{x}_{ij}$ by making them look as realistic as possible to fool the discriminator.

Writer Classification Loss. Following the same architecture of D we propose to make use of a writer classifier W to distinguish which is the author of a given word image. The writer classification loss is the cross entropy loss, commonly used for classification systems. This component is responsible for guiding our model on properly transferring the desired calligraphic style from x_j to the generated image.

Content Recognition Loss. Taking the same architecture as the main HWR which is overviewed within the red box in Figure 1, we implement yet another separate HWR system R to lead our generation process towards the desired textual content provided by F_c. The loss of the HWR R is the same as our main HWR defined in Equation 4. This last component is responsible to guarantee that the generated image actually conveys the same textual content than the input image x_i.

The joint training process follows as a min-max game where all the described losses are combined in a weighted manner. In this end-to-end model, we set two weights γ_1, γ_2 for the main recognition loss in the red box of Figure 1 and the summation of three auxiliary losses of disentanglement module in the blue box, respectively. Note that the min-max problem comes from the discriminator part, which follows the traditional training models for adversarial approaches.

Finally, our implementation details are the following. Our experiments were run using PyTorch [29] on a cluster of NVIDIA GPUs. Source code is available [1]. The training was done using the Adam optimizer with an initial learning rate of $2 \cdot 10^{-4}$ and a batch size of 8. We have set the dropout probability to be 50% for all the GRU layers. The training set is shuffled at each epoch and the input image pair (x_i, x_j) is also chosen randomly among all the writers in the training set. During the training, the weights γ_1 and γ_2 are both set to 1, then after reaching a specific epoch number, γ_2 should be gradually decreased until 0. To further boost the HWR performance, fine-tuning can be applied on the main recognition module (red box of Figure 1) with IAM training data.

IV. Experimental Evaluation

A. Dataset and Performance Evaluation

To carry our experiments, we have chosen the IAM offline handwriting dataset [30], being one of the most popular and widely used benchmarks in the field of handwriting recognition. We have used the RWTH Aachen partition for the dataset, composed of 55,081, 8,895 and 25,920 word images for training, validation and test sets respectively. Furthermore, the IAM dataset provides not only text images and their corresponding transcriptions, but also the writer identifier. Based on the assumption that each writer has one specific writing style, we have 500 handwriting styles in the IAM dataset, with 283 writers in the training set, 56 writers for validation and 161 writers in the test set.

In order to evaluate the effectiveness on the recognition performance, we will use the standard error measures at character and word level. The Character Error Rate (CER) and the Word Error Rate (WER) [31], are computed as the Levenshtein distance at either character or word level. Since we focus our experiments on individual words, the WER measure is the inverse of the overall word accuracy.

B. Qualitative Evaluation of the Generative Process

First and foremost, in order to validate that the proposed method is able to really learn properly disentangled feature representations for both contents and writing styles, from the content and style encoders C and S, we will qualitatively analyze the generative part of our approach.

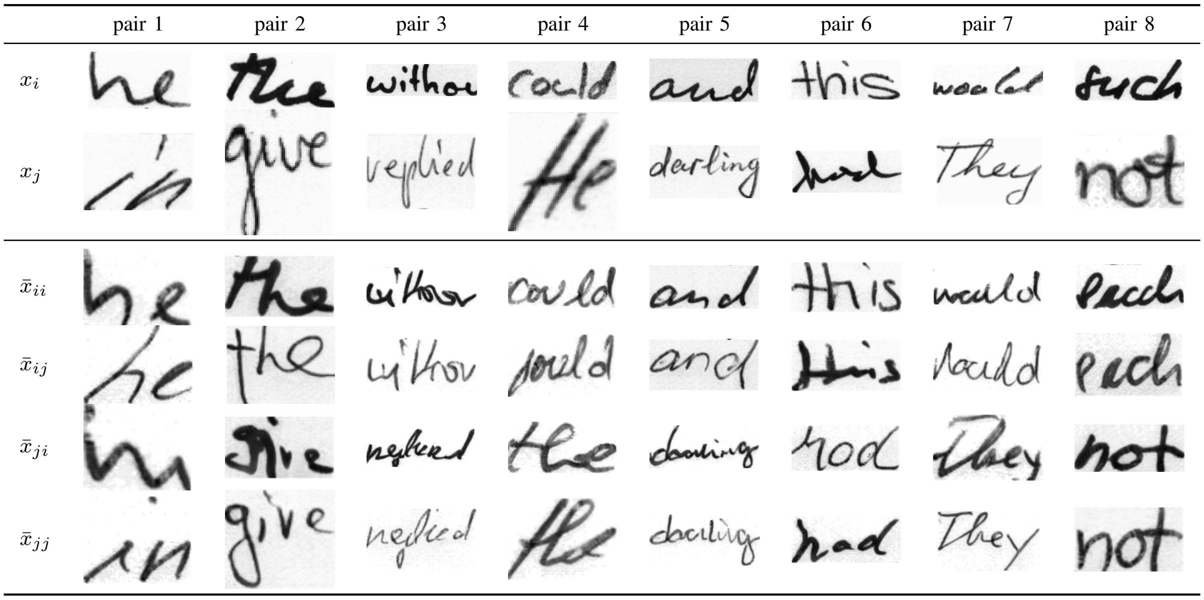
Given a pair of input word images x_i and x_j, four different permutations, style and content-wise, are possible to be generated. Formally, the generated images $\bar{x}_{ii}$, $\bar{x}_{ij}$, $\bar{x}_{ji}$ and $\bar{x}_{jj}$ will be the result of using either x_i or x_j as input for the content or style encoders (*c.f.* equation 2). We show in Table I some results of such generative process. Images $\bar{x}_{ii}$ and $\bar{x}_{jj}$ shall be the re-writings with the same content and style of x_i and x_j respectively, while $\bar{x}_{ij}$ and $\bar{x}_{ji}$ shall correspond to images conveying the content of x_i while appearing to be written with the style of x_j, and vice-versa. In order to properly transfer both style and content within the generator, the feature tensors F_c and F_s should be completely disentangled, which means that the content representation F_c should just encode the textual features that conform a certain word completely disregarding the writing style. We observe in such sample results how effectively the learned content and style feature representations have been properly disentangled one from each other. The qualitative results from the generative part, although not being the ultimate goal of this work, are encouraging enough to think that the content features F_c shall be at a certain extent agnostic to the handwriting styles.

C. Handwriting Recognition Performance

In order to quantitatively evaluate the performance of the proposed style invariant content features within the recognition pipeline, we present in Table II some comparative results. In this experiment we have trained exactly the same sequence-to-sequence neural architecture in three different setups that are

[1] https://github.com/omni-us/research-ContentDistillation-HTR

TABLE I
QUALITATIVE RESULTS OF THE CONTENT AND STYLE GUIDED GENERATIVE PROCESS. GIVEN A PAIR OF INPUT IMAGES, WE ARE ABLE TO GENERATE THE FOUR DIFFERENT PERMUTATIONS OF STYLES AND TEXTUAL CONTENTS.



objectively comparable. On the one hand, we just use the IAM training set images with data augmentation, and we reach a CER value of 6.88%. To push that value forward, we make use of not only IAM training set, but also unlabelled IAM test set to do a domain adaptation between both sets by using the adversarial domain adaptation technique proposed in [32], so that the feature distribution of test set samples shall be properly adapted to that of training samples. We observe that the yielded CER is of 6.75% in that case. We finally jointly train the recognizer with the generative pipeline in an end-to-end fashion, and, we just use IAM training samples, without any other additional image. We observe that the obtained error rates are lower than the two previous approaches, 6.43% and 16.39% respectively, indicating that the learned features are actually better focused to the conveyed textual contents, while being resilient to handwriting style changes.

TABLE II
RECOGNITION PERFORMANCE FOR SEQUENCE-TO-SEQUENCE NETWORK WITH THREE DIFFERENT TRAINING STRATEGIES.

Training Procedure	CER (%)	WER (%)
IAM Training Set [11]	6.88	17.45
Domain Adaptation [17]	6.75	17.26
Content Distillation (Proposed)	**6.43**	**16.39**

D. Comparison with State of the Art

Finally, in order to put in context the previous performance evaluation measures, we provide in Table III a comparison with the the state-of-the-art methods for handwritten word recognition. To give a fair comparison, we have just selected works focused at word level, and from them, we report the error rates from those models that do not entail any language model nor closed lexicon. We observe that our proposed approach compares quite satisfactorily with the rest of the state-of-the-art methods. The exception is the approach of Dutta *et al.* [33], however this work provides their results on IAM by pre-training on 9M synthetic data of IIIT-HWS [34], de-slanting word images as pre-processing, and using test-time augmentation, which make such results not directly comparable with the rest of the reported error measures.

TABLE III
COMPARISON WITH THE STATE-OF-THE-ART METHODS.

Approach	Method	CER (%)	WER (%)
RNN + CTC	Mor *et al.* [35]	–	20.49
	Pham *et al.* [36]	13.92	31.48
	Krishnan *et al.* [37]	6.34	16.19
	Wiginton *et al.* [8]	6.07	19.07
Seq2Seq + Attention	Bluche *et al.* [27]	12.60	–
	Sueiras *et al.* [38]	8.80	23.80
	Zhang *et al.* [16]	8.50	22.20
	Dutta *et al.* [33]	4.88	12.61
	Proposed	6.43	16.39

V. CONCLUSION

In this paper we have presented a novel training approach for handwritten word recognition that is able to disentangle the content and style features of input images. The proposed method jointly optimizes a generative process and a handwritten word recognizer with the aim of yielding a style-independent content-centric feature representation that boosts the recognition performance in multi-writer scenarios. The presented results prove that by coupling a generative and a recognition process we are able to achieve separated content and calligraphic stylistic features that serve both to style and content transfer and for better handwritten word recognition.

Acknowledgment

This work has been partially supported by the EU Commission H2020 SME Instrument project 849628, the grant 2016-DI-087 from the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya, the Spanish projects TIN2017-89779-P and RTI2018-095645-B-C21, and the grants FPU15/06264 and RYC-2014-16831. The Titan XP used for this research were donated by NVIDIA.

References

[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[2] A.-L. Bianne-Bernard, F. Menasri, R. A.-H. Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and contextual information in HMM modeling for handwritten word recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2066–2080, 2011.

[3] S. España-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martínez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 767–779, 2010.

[4] A. Giménez, I. Khoury, J. Andrés-Ferrer, and A. Juan, "Handwriting word recognition using windowed Bernoulli HMMs," *Pattern Recognition Letters*, vol. 35, pp. 149–156, 2014.

[5] 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, 2008.

[6] P. Krishnan, K. Dutta, and C. Jawahar, "Word spotting and recognition using deep embedding," in *Proceedings of the IAPR International Workshop on Document Analysis Systems*, 2018.

[7] B. Stuner, C. Chatelain, and T. Paquet, "Handwriting recognition using cohort of LSTM and lexicon verification with extremely large lexicon," *arXiv preprint arXiv:1612.07528*, 2016.

[8] C. Wigington, S. Stewart, B. Davis, B. Barrett, B. Price, and S. Cohen, "Data augmentation for recognition of handwritten words and lines using a CNN-LSTM network," in *Proceedings of the IAPR International Conference on Document Analysis and Recognition*, 2017.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014.

[11] L. Kang, J. I. Toledo, P. Riba, M. Villegas, A. Fornés, and M. Rusiñol, "Convolve, attend and spell: An attention-based sequence-to-sequence model for handwritten word recognition," in *Proceedings of the German Conference on Pattern Recognition*, 2018.

[12] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, "Evaluating sequence-to-sequence models for handwritten text recognition," *arXiv preprint arXiv:1903.07377*, 2019.

[13] A. K. Bhunia, A. Das, A. K. Bhunia, P. S. R. Kishore, and P. P. Roy, "Handwriting recognition in low-resource scripts using adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[14] N. Gurjar, S. Sudholt, and G. A. Fink, "Learning deep representations for word spotting under weak supervision," in *Proceedings of the IAPR International Workshop on Document Analysis Systems*, 2018.

[15] P. Krishnan and C. Jawahar, "HWNet v2: An efficient word image representation for handwritten documents," *International Journal on Document Analysis and Recognition*, vol. 22, no. 4, pp. 387–405, 2019.

[16] Y. Zhang, S. Nie, W. Liu, X. Xu, D. Zhang, and H. T. Shen, "Sequence-to-sequence domain adaptation network for robust text image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[17] L. Kang, M. Rusiñol, A. Fornés, P. Riba, and M. Villegas, "Unsupervised adaptation for synthetic-to-real handwritten word recognition," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2020.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014.

[19] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "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.

[20] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[21] R. Gomez, A. F. Biten, L. Gomez, J. Gibert, D. Karatzas, and M. Rusiñol, "Selective style transfer for text," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2019.

[22] J. Li, S. Wang, Y. Wang, and Z. Tang, "Synthesizing data for text recognition with style transfer," *Multimedia Tools and Applications*, vol. 78, no. 20, pp. 29 183–29 196, 2019.

[23] Y. Wang, Z. Lian, Y. Tang, and J. Xiao, "Boosting scene character recognition by learning canonical forms of glyphs," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, no. 3, pp. 209–219, 2019.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision*, 2018.

[26] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[27] T. Bluche, J. Louradour, and R. Messina, "Scan, attend and read: End-to-end handwritten paragraph recognition with MDLSTM attention," in *Proceedings of the IAPR International Conference on Document Analysis and Recognition*, 2017.

[28] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proceedings of the Advances in Neural Information Processing Systems*, 2015.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[30] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[31] V. Frinken and H. Bunke, "Continuous handwritten script recognition," in *Handbook of Document Image Processing and Recognition*, 2014, pp. 391–425.

[32] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495*, 2014.

[33] K. Dutta, P. Krishnan, M. Mathew, and C. Jawahar, "Improving CNN-RNN hybrid networks for handwriting recognition," in *Proceedings of the International Conference on Frontiers in Handwriting Recognition*, 2018.

[34] P. Krishnan and C. Jawahar, "Generating synthetic data for text recognition," *arXiv preprint arXiv:1608.04224*, 2016.

[35] N. Mor and L. Wolf, "Confidence prediction for lexicon-free OCR," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2018.

[36] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *Proceedings of the International Conference on Frontiers in Handwriting Recognition*, 2014.

[37] P. Krishnan, K. Dutta, and C. Jawahar, "Word spotting and recognition using deep embedding," in *Proceedings of the IAPR International Workshop on Document Analysis*, 2018.

[38] J. Sueiras, V. Ruiz, A. Sanchez, and J. F. Velez, "Offline continuous handwriting recognition using sequence to sequence neural networks," *Neurocomputing*, vol. 289, pp. 119–128, 2018.