DEFORMATION-INVARIANT NETWORKS FOR HANDWRITTEN TEXT RECOGNITION

George Retsinas[1] *Giorgos Sfikas*[2] *Christophoros Nikou*[2] *Petros Maragos*[1]

[1]School of E.C.E., National Technical University of Athens, 15773, Athens, Greece
[2]Dept. of Computer Science & Engineering, University of Ioannina, 45110, Ioannina, Greece
Email: gretsinas@central.ntua.gr, sfikas@cs.uoi.gr, cnikou@cs.uoi.gr, maragos@cs.ntua.gr

2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506414

ABSTRACT

Image deformations under simple geometric restrictions are crucial for Handwriting Text Recognition (HTR), since different writing styles can be viewed as simple geometrical deformations of the same textual elements. In this respect, the usefulness of including deformation invariance to an HTR system is indisputable. We explore different existing strategies for ensuring deformation invariance, including spatial transformers and deformable convolutions, under the context of text recognition, as well as introduce a new deformation-based algorithm, inspired by adversarial learning, which aims to reduce character output uncertainty during evaluation time. The resulting HTR system is shown to achieve state-of-the-art performance on the IAM and RIMES datasets.

***Index Terms*—** Handwritten Text Recognition, Deformable Convolution, Spatial Transformer Network

1. INTRODUCTION

Handwritten Text Recognition (HTR) is one of the key tasks in the domain of document image processing. Unlike recognition of machine-printed text, handwriting is related to a number of unique characteristics that make the task much more challenging than traditional optical character recognition (OCR). The most salient of these characteristics is potentially high writing variability between writers or even within the text of the same writer. This translates to a significant challenge to design an effective, generalizable learning system, with transferability between different learned writing styles more often not being a given [1]. Neural networks (NNs), among a variety of other learning systems, have been used for the recognition of handwriting from early on, with a span ranging between simpler subtasks such as single digit recognition [2] up to full, unconstrained offline HTR. The seminal work of Graves et al. [3] enabled neural net training without assuming any prior character segmentation, inspiring a plethora of subsequent works for HTR. As it is desirable to supply NNs, and especially their more modern variants with as much data as possible, data augmentation techniques have been employed as a standard part of the training pipeline. Augmentation has been shown to be a valuable tool that can aid a model to generalize better. Regarding HTR in particular, augmentation strategies include applying small linear or non-linear deformations on existing training images [2]. This way, writing the same characters or words with a different style or manner may be simulated.

This research has been partially co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call OPEN INNOVATION IN CULTURE, project *Bessarion* (T6YBΠ-00214).

Related to exploiting the role of deformations as an elementary operation in the context described above, deformable convolutions and spatial transformers are two concepts that were relatively recently proposed in the context of NN architectures [4, 5]. Deformable convolutions generalize the concept of convolution by redefining the form of the convolution as being of an adapatable shape. Convolution weights are expected to multiply inputs not on the usual orthogonal, canonical $K \times K$ grid, but the weight-input coordinate correspondence becomes subject to learning. Spatial transformers are NN modules that apply a learnable deformation on the input image or feature map. Subsequent layers are then expected to process the transformed input more efficiently; for example, an object may be transformed to a pose that is more convenient for inference in the next layers. Both deformable convolution and spatial transformer parameters can be learned using standard back-propagation.

While modules and concepts that are based on the notion of applying a deformation on inputs, feature maps or input coordinate frames such as the ones described above have been succesful in numerous vision tasks, in HTR their use is still far from being fully exploited or even explored. With this paper, we aim to explore the potential of the aforementioned concepts in the context of the HTR task. Furthermore, we propose a novel adversarial-based method that can be easily adapted to vision tasks other than HTR. In particular, the contribution of this work is two-fold: a) We use deformation-invariance imposing strategies in the context of HTR [1]. b) We propose a novel algorithm that leads to increased HTR performance. Specifically, given a trained network, we decide if a small deformation can improve the overall score of the network by reducing uncertainty over the output probabilities. The algorithm relies on the rationale of generating and training with adversarial examples [7].

In section 2, we focus on the formulation of handwritten text deformations, while in section 3, we describe in detail the backbone of our system, a compact convolutional-only NN, along with the proposed deformation-invariant modules built for our HTR system. In order to further increase performance, in section 4, we introduce an unsupervised algorithm which iteratively reduces output uncertainty during evaluation. The proposed methods are evaluated with numerical experiments in section 5. We close the paper with a conclusion and discussion of future work in section 6.

2. DEFORMATION INVARIANCE IN HTR MODELS

In general, learning machines typically are expected to provide an unchanged output given *some kind* of transformation of the input variables. The exact type and magnitude of transformation to which the learning machine is expected to be invariant, largely depends on

[1]Publised after this work had been submitted, [6] use deformable convolution for HTR.

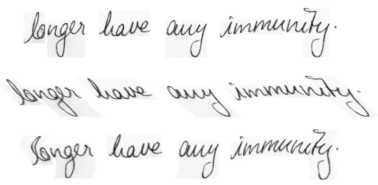

Fig. 1: Examples of text image deformations. Top: initial text image. Middle: Global affine transformation. Bottom: Local deformation.

the specifics of the given task. For example, we require a vision system to be able to detect an object regardless of its size and position in the image, hence in this respect translation and scale invariance would be desired traits of the detector. The magnitude of a transformation may also be relevant: In speech recognition, nonlinear warping of a small magnitude along the time axis that preserves temporal ordering should leave signal interpretation unchanged. Likewise for HTR, it is desirable to obtain the same recognition output under specific types and magnitudes of transformation families.

We distinguish two categories of relevant text deformations (see Fig. 1 for visual examples): (a) Global affine: Basic geometric transformation over the entire image which includes translation, scale, rotation and slant/skew. The latter corresponds to a very common variation in writing style. (b) Small local deformations: Non-uniform geometric deformations, which affect differently each part of the image, can be represented by a grid of coordinate offset values. This deformation is often referred to as elastic net deformation. Note that global affine deformation is a special case of the local deformation formulation. However, in the context of developing a trainable module the former can be trained considerably easier due to limited number of parameters (6 parameters for the affine case).

Two major strategies exist in order to implement transformation invariance for NN-based HTR models: *Augmentation*, which adds transformed replicas of original data according to the required invariances, and ensuring *Network structural invariance*, according to which invariance properties are built into the structure of the network. We focus on the latter, assuming the former as a de facto strategy in all modern HTR systems. Other strategies are also possible [2], which are however out of the scope of the present work.

Augmentation: Augmentation [2] is the strategy according to which new samples are generated on the basis of the existing data. Augmentation has proven beneficial to a vast variety of tasks in deep learning. In classical augmentation methods, random transformations are applied to the original data [8]. A more modern approach employs Generative Adversarial Networks for augmentation, which has been experimented with successfully in a number of tasks [9, 10]. One promising and rising field of research is adversarial augmentation [7], which seeks "hard" adversarial examples, forcing the model to generate more precise classification boundaries. Augmentation by default refers to adding data to the training set, however it can also refer to test-time augmentation, where test data are transformed in some way in order to obtain more accurate inference. These research directions –adversarial learning and test-time augmetation– are closely related to the algorithm we propose in Section 4.

Network structural invariance: According to this strategy, invariance is to be built into the network structure. Two of the most succesful components to handle global and local deformations are deformable convolutions [4] and Spatial Transformer Networks [5].

Deformable convolutions generalize the well-known operation of convolution in a straightforward manner, by adapting the neighbourhood information as a function of a trainable offset map. In particular, deformable convolutions are defined as:

$$g(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) f(p_0 - p_n + \Delta p_n) \tag{1}$$

where $\mathcal{R}$ is a set of $2D$ offsets that form a canonical grid around the point p_0 for which we compute convolution. Offsets Δp_n can be and typically are fractional, and in conjunction with corresponding p_n offsets, they result in sampling input feature map f on non-canonical positions. Most importantly, offsets Δp_n are trainable. After sampling f, the feature map is multiplied by convolution weights and summed to provide output g, as in standard convolution.

The main idea of spatial transformers is also intuitive, and similar to the idea behind deformable convolutions: jointly optimize the inference module along with a normalization module. The later will provide such features that will assist the former. Given a set of constraints over the transformation, e.g. global affine, one can design a module that generates an offset grid that complies to these restrictions and can transform the input image to a deformed one according to the offset map using interpolation. The offset grid is the output of a trainable module, with a set of parameters to be optimized. It is expected that spatial transformer modules will reduce the variability of the input images with respect to the aforementioned constraints. A typical paradigm is the use of an affine transformation, meaning that the STN module estimates 6 affine transformation parameters and then transform accordingly the image. Spatial transformers can be used on the input image, as well as intermittently between network layers in order to normalize intermediate feature maps. Formally, a spatial transformer module is defined as:

$$g(p_0) = \sum_{p_n \in \mathcal{R}} f(p_n) k_{p_n}(p_0) \tag{2}$$

where k corresponds to a sampling kernel. For example, the kernel $k_m(x) = \delta(\lfloor x + 0.5 \rfloor - m)$ where δ is the Dirac delta function, corresponds to nearest neighbour interpolation. In practice, bilinear interpolation is preferred.

3. PROPOSED NETWORK ARCHITECTURE

In order to explore the impact of deformations we rely on deploying an efficient yet compact neural network as reference, capable of attaining competitive results without any modifications. Contrary to the majority of contemporary HTR nets which use recurrent nets and especially LSTM blocks [11, 12, 8], we propose a convolutional-only system. We have opted to do so for two reasons: First, our focus in this work concerns exploring the uses of deformation-related operations. Second, 1D convolutional layers can also model complex contextual information and address HTR efficiently [13] with only a fraction of their computational cost.

The proposed architecture consists of two components: the *Convolutional Backbone*, which transforms the input image into a feature map through consecutive ResNet blocks [14], and the *Convolutional Head*, which takes as input a flattened feature map and translates it into character predictions using $1D$ convolutions. Applying the softmax function on the final output, we form a sequence of probability distributions over the possible characters which is then propagated into a Connectionist Temporal Classification (CTC) loss [3]. Decoding of the output during evaluation is performed by a greedy selection of the characters with the highest probability at each step. The network architecture is summarized in Figure 2a. The flattening operation between the backbone and the head is column-wise max-pooling, reducing the vertical information on the output feature map into a single $1D$ feature vector per column. The convolutional head then acts as a temporal encoder which combines neighboring information, adding gradually more context with each processing layer.

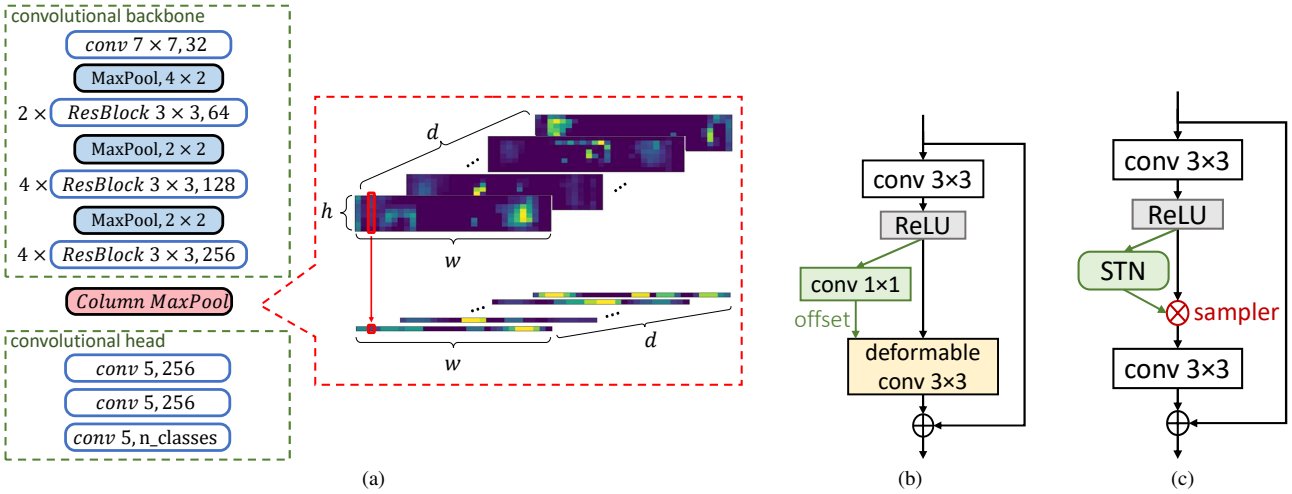

Fig. 2: (a) Proposed CNN architecture consisted of Convolutional Backbone, Column MaxPooling and Convolutional Head (1D Convs). (b) ResBlock variation for deformable convolutions. (c) ResBlock variation for the STN case.

Note that the proposed architecture is inherently translation invariant: the CTC logic provides horizontal translation invariance, while the pooling operation after the CNN backbone, which produces a sequence of features, provides vertical translation invariance.

Concerning modules to ensure structural invariance, we integrate them to the architecture as follows. Deformable convolutions are injected directly into the architecture, taking the place of standard convolution operations in the ResNet blocks (ResBlock), as shown in Figure 2b. Concerning STN modules, we have found that we can get excellent results by using a spatial transformer model as a preprocessor before the vanilla backbone. However, we have also experimented injecting a shallow STN into the ResNet block between the two convolutions, as shown in Figure 2c. We further discuss use of STNs and deformable convolutions in tandem in Section 5.

4. PROPOSED UNCERTAINTY REDUCTION METHOD

In this section we present a new method for reinforcing deformation-invariance of a trained HTR network, that is applicable at test time. Conceptually, we borrow from the field of adversarial learning, where a basic operation is to propagate through a pre-trained network so as to construct new data that completely fail and mislead the network to wrong predictions.

In our method, we seek to improve the predictions without any prior knowledge over the labels. For the case of HTR, the NN final layer output is a sequence of probabilities over the possible set of characters. Instances of misclassification under this paradigm are closely related to highly uncertain predictions, where two or more characters have high probability scores. That is not to say that a probabilistic model formulation is responsible for these errors, but rather that it is more unlikely that a correct recognition result will be obtained when our per-character predictions are uncertain. This observation leads us to an intuitive formulation of a method: we aim to find a small deformation that maximizes the overall probability of the predicted characters, therefore minimizing output uncertainty.

We formulate our method, to which we shall refer to as Uncertainty Reduction (UR), as follows. Let us denote our (trained) NN as $\mathbf{Y} = \phi(\mathbf{X})$, which takes as input an image $\mathbf{X}$ and returns a sequence of softmax outputs $\mathbf{Y}$ of size $w \times n$, where w is the length of the sequence and n the number of possible characters. Having more confident results is equivalent to having as many per-character softmax predictions as possible that are each close to having one class close to a probability of 1. Formally, we write this objective as $\sum_{i=1}^{w}\sum_{j=1}^{n} y_{ij}^2$, where $\mathbf{Y} = [y]_{ij}$. It is straightforward that this objective can be shorthanded as $||\mathbf{Y}||_F^2 = tr(\mathbf{Y}^\intercal\mathbf{Y})$ where $||\cdot||_F$ is the Frobenius norm. This objective is cast as a loss function $\mathcal{L}$:

$$\mathcal{L}(\mathbf{Y}) = -||\phi(\mathbf{X})||_F, \tag{3}$$

where in contrast to standard training we now seek to optimize $\mathcal{L}$ w.r.t. test image X and assuming network weights to be "frozen". The minimal value of this loss is attained when there is no uncertainty at each sequence step, i.e. $0 < -\mathcal{L} < w$.

In practice this optimization task is under constraints; we are interested in deformations of a small magnitude over the test input - initial point X. Unconstrained optimization is to be avoided, as if we allowed large deformations then the solution would tend to coincide with extreme, unusable transformations. For example, such a result would be an overly shrinked input, in order to predict with high certainty the space character ignoring the existing characters, since the algorithm is agnostic to the true labels of the text. Starting from input X, we need to find small local deformations of the image $\mathcal{T}(\mathbf{I}, \mathbf{d})$, i.e. find an appropriate grid offset $\mathbf{d}$ of small magnitude, where $\mathcal{T}$ denotes a bilinear grid sampler. In order to meet these criteria, we opt to use a small number k of gradient descent steps. Formally, we use the following iterative formula to optimize deformation $\mathbf{d}$:

$$\mathbf{d}_{i+1} = \mathbf{d}_i - \eta\nabla\mathcal{L}(\phi(\mathcal{T}(\mathbf{I}, \mathbf{d}_i))) \tag{4}$$

5. EXPERIMENTAL EVALUATION

Evaluation of our methods is performed on two widely used datasets, IAM [15] and Rimes [16]. All experiments follow the same setting: line-level recognition, training for 240 epochs with cosine annealing and restarts every 40 epochs [17] and evaluating using a lexicon-free greedy CTC decoding scheme. The pre-processing steps are: 1) All images are resized to a resolution of 128×1024 pixels. Initial images are padded (using the image median value, usually zero) in order to attain the aforementioned fixed size. If the initial image is greater than the predefined size, the image is rescaled.

The padding option aims to preserve the existing aspect ratio of the text images. 2) A simple global affine augmentation is applied at every image, considering only rotation and skew of small magnitude in order to generate valid images. Additionally local deformations and morphological operations, based on max-pooling, are applied as in [12]. 3) Each line transcription has spaces added before and after, i.e. "He rose from" is changed to " He rose from ". This operation aims to assist the system to predict the marginal spaces that exist in the majority of the images. Experiments were run on the IAM dataset unless explicitly stated otherwise. Character Error Rate (CER) and Word Error Rate (WER) metrics are reported in all cases (lower values are better). Code is available at `https://github.com/georgeretsi/defHTR`.

Using and combining structural Deformation Invariance Modules: First, we evaluate different settings of spatial transformers. Global affine parameters are reduced to 4 instead of the maximum of 6. We exclude translation parameters since the proposed architecture is already translation-invariant as explained in Section 3. We distinguish two granularities of deformations, global (affine) or local, along with two variations of including STN into the network, at input level (consisted of 4 ResBlocks) or at ResBlock level (multiple STNs that use 1×1 convolutions). The results are summarized in Table 1a. STNs based on global deformation under small affine transfomations can be helpful regardless of their usage, i.e. input or ResBlock. On the other hand, we observe that the elastic net variant performs worse than the CNN baseline for both alternatives. In practice, the training of such local STNs has proven to be difficult and they do not converge into helpful solutions with typical unconstrained back propagation.

Def.	Variation	CER%	WER%
baseline		5.10	18.03
Global	Input	4.88	17.76
Global	ResBlock	4.94	17.65
Local	Input	5.01	19.01
Local	ResBlock	6.12	20.25

(a)

Level	CER%	WER%
baseline	5.10	18.03
0	5.12	18.00
1	4.89	17.63
2	4.91	17.07
1, 2, 3	4.67	16.57

(b)

Table 1: Exploration of different architectural settings for (a) STNs and (b) deformable convolutions.

Next, we focus on deformable convolutions. In Table 1b, we explore the impact of introducing deformable convolutions at different levels of the architecture (the ResBlocks between downsamplings correspond to a level, see Fig. 2a). With the exception of the first level, introducing a deformable ResBlock improves the overall performance, incrementally at each level, while adding deformable convolutions along the entire architecture considerably outperforms the baseline network and the STN variants. This result supports the theoretical advantage of deformable convolutions over STNs: they are more flexible, since a different local deformation can be defined as fine as the pixel level.

Additionally, we examine the case of simultaneously adding an affine STN at the input of the network, along with the modified deformable architecture. The results were identical, with or without the STN module, hinting that deformable convolutions produce invariant features that include the STN normalization.

Uncertainty Reduction method: Finally, we validate the *UR* algorithm we proposed in section 4.The results are summarized in Table 2, considering several variations. Specifically, *CNN* and *defCNN* denote the baseline and its variation of deformable convolutions, respectively, while *STN* denote the global affine input version of the STN implementation.

The results highlight the capacity of improvement of HTR networks through minor deformations. In other words, the minimization of uncertainty assists to uncover more reliable solutions close to the initial one. As expected, larger margin for improvement is observed when the system is underperfoming, e.g. the initial CNN baseline. Nonetheless, the proposed algorithm consistently improves performance regardless of the choice of the architecture. This performance increase comes at the cost of extra execution time requirements, since we should back-propagate through the network several times ($k = 5$ for the reported results).

	w/o UR		w/ UR	
Architecture	CER %	WER %	CER %	WER
CNN	5.10	18.03	4.95	17.31
CNN + STN	4.88	17.76	4.69	16.89
defCNN	4.67	16.57	4.55	16.08
defCNN + STN	4.66	16.63	4.55	16.25

Table 2: Impact of the proposed Uncertainty Reduction algorithm for $k = 5$ iterations.

State-of-the-Art Comparison: In order to grasp the significance of the reported performance, we compare the proposed methods with the existing literature (lexicon-free line-level HTR). The results are summarized in Table 3. For the case of IAM dataset, we considerably outperform every reported method, including approaches based on LSTMs [12], attention mechanisms [18] and Sequence-to-Sequence transation [19, 20]. Considering RIMES dataset, we also achieve SoA results, without outperforming all existing methods.

	IAM		RIMES	
Architecture	CER %	WER %	CER %	WER %
Chen et al. [21]	11.15	34.55	8.29	30.5
Pham et al. [22]	10.8	35.1	6.8	28.5
Khrishnan et al. [23]	9.78	32.89	-	-
Chowdhury et al. [19]	8.10	16.70	3.59	9.60
Puigcerver [12]	6.2	20.2	2.6	10.7
Markou et al. [24]	6.14	20.04	3.34	11.23
Dutta et al. [8]	5.8	17.8	5.07	14.7
Tassopoulou et al. [25]	5.18	17.68	-	-
Yousef et al. [18]	4.9	-	-	-
Michael et al. [20]	4.87	-	-	-
Cojocaru et al. [6]	4.6	19.3	4.6	14.8
CNN	5.10	18.03	3.43	12.78
CNN + UR	4.95	17.31	3.33	12.25
defCNN	4.67	16.57	3.11	11.05
defCNN + UR	4.55	16.08	3.04	10.56

Table 3: Performance comparison for IAM/RIMES datasets.

6. CONCLUSION

In this work, we proposed a convolutional-only HTR architecture that uses techniques for deformation invariance in order to boost performance. To further increase HTR performance, we proposed an iterative unsupervised algorithm for reducing the character uncertainty at the softmax output of the network. Considering future work, the experimental exploration hints towards several possible extensions: explore thoroughly the training strategy of the local deformation STN variant, apply the proposed approaches in conjunction with a RNN and formulate an appropriate adversarial augmentation scheme, akin to the methodology presented in Section 4, which aims to maximize uncertainty as a generator of "hard" examples.

7. REFERENCES

[1] George Retsinas, Giorgos Sfikas, and Basilis Gatos, "Transferable deep features for keyword spotting," in *Multidisciplinary Digital Publishing Institute Proceedings*, 2018, vol. 2, p. 89.

[2] Christopher M Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[3] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369–376.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 764–773.

[5] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., "Spatial transformer networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 2017–2025, 2015.

[6] Iulian Cojocaru, Silvia Cascianelli, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara, "Watch your strokes: Improving handwritten text recognition with deformable convolutions," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 6096–6103.

[7] Chih-Hui Ho and Nuno Vasconcelos, "Contrastive learning with adversarial examples," *arXiv preprint arXiv:2010.12050*, 2020.

[8] Kartik Dutta, Praveen Krishnan, Minesh Mathew, and CV Jawahar, "Improving CNN-RNN hybrid networks for handwriting recognition," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 80–85.

[9] Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger Gunn, Alexander Hammers, David Alexander Dickie, Maria Valdés Hernández, Joanna Wardlaw, and Daniel Rueckert, "GAN augmentation: Augmenting training data using generative adversarial networks," *arXiv preprint arXiv:1810.10863*, 2018.

[10] Panagiotis Dimitrakopoulos, Giorgos Sfikas, and Christophoros Nikou, "ISING-GAN: Annotated data augmentation with a spatially constrained generative adversarial network," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1600–1603.

[11] Paul Voigtlaender, Patrick Doetsch, and Hermann Ney, "Handwriting recognition with large multidimensional long short-term memory recurrent neural networks," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2016, pp. 228–233.

[12] Joan Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2017, vol. 1, pp. 67–72.

[13] George Retsinas, Giorgos Sfikas, and Petros Maragos, "WS-RNet: Joint spotting and recognition of handwritten words," *arXiv preprint arXiv:2008.07109*, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[15] U-V Marti and Horst 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.

[16] Emmanuèle Grosicki, Matthieu Carre, Jean-Marie Brodin, and Edouard Geoffrois, "RIMES evaluation campaign for handwritten mail processing," 2008.

[17] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[18] Mohamed Yousef, Khaled F Hussain, and Usama S Mohammed, "Accurate, data-efficient, unconstrained text recognition with convolutional neural networks," *Pattern Recognition*, vol. 108, pp. 107482, 2020.

[19] Arindam Chowdhury and Lovekesh Vig, "An efficient end-to-end neural model for handwritten text recognition," 2018.

[20] Johannes Michael, Roger Labahn, Tobias Grüning, and Jochen Zöllner, "Evaluating sequence-to-sequence models for handwritten text recognition," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1286–1293.

[21] Zhuo Chen, Yichao Wu, Fei Yin, and Cheng-Lin Liu, "Simultaneous script identification and handwriting recognition via multi-task learning of recurrent neural networks," in *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2017, vol. 1, pp. 525–530.

[22] Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *2014 14th international conference on frontiers in handwriting recognition*. IEEE, 2014, pp. 285–290.

[23] Praveen Krishnan, Kartik Dutta, and CV Jawahar, "Word spotting and recognition using deep embedding," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 1–6.

[24] K Markou, L Tsochatzidis, K Zagoris, A Papazoglou, BX Karagiannis, S Symeonidis, and I Pratikakis, "A convolutional recurrent neural network for the handwritten text recognition of historical greek manuscripts," in *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10-15, 2021, Proceedings, Part VII*. Springer International Publishing, 2021, pp. 249–262.

[25] Vasiliki Tassopoulou, George Retsinas, and Petros Maragos, "Enhancing handwritten text recognition with n-gram sequence decomposition and multitask learning," *ICPR*, 2021.