





Exploiting Context in Handwriting Recognition Using Trainable Relaxation Labeling

Sara Ferro^{2,1,*}^a, Alessandro Torcinovich^{3,*}^b, Arianna Traviglia¹^c and Marcello Pelillo^{2,1}^d

¹Center for Cultural Heritage Technology, Italian Institute of Technology, Venice, Italy

²DAIS, Ca' Foscari University of Venice, Venice, Italy

³Department of Computer Science, ETH Zürich, Zürich, Switzerland

Keywords: Handwriting Recognition, Relaxation Labeling Processes, Generalisation.


Abstract: Handwriting Text Recognition (HTR) is a fast-moving research topic in computer vision and machine learning domains. Many models have been introduced over the years, one of the most well-established ones being the Convolutional Recurrent Neural Network (CRNN), which combines convolutional feature extraction with recurrent processing of the visual embeddings. Such a model, however, presents some limitations such as a limited capability to account for contextual information. To counter this problem, we propose a new learning module built on top of the convolutional part of a classical CRNN model, derived from the relaxation labeling processes, which is able to exploit the global context reducing the local ambiguities and increasing the global consistency of the prediction. Experiments performed on three well-known handwritten recognition datasets demonstrate that the relaxation labeling procedures improve the overall transcription accuracy at both character and word levels.


1 INTRODUCTION


The increased availability of digitized large collections of ancient and historical manuscripts in the form of digital images is increasingly providing the opportunity to convert past scripts into machine-readable text formats and create digital storages of editable content. Advancements both in layout analysis and text recognition are currently paving the way for making digitized manuscript content fully machine searchable and readable. Automatic letter/word recognition of ancient and historic texts, however, still has many inherent complexities. Besides the broad range of handwriting across several centuries, there are also many varieties in the shape of the letters due to the writing tool type and width within the same group of scripts; in addition, individual scribes might have had different handwriting (e.g. in letters shape and size), this being a common feature, especially


for historical documents. Transcription of handwritten text from badly preserved documents is also made more difficult by the deterioration of the writing support (paper, parchment or others).

Even though, at present, handwriting transcription has shown promising performances (Teslya and Mohammed, 2022), (Lombardi and Marinai, 2020), thanks to both the development of sequential models which are compatible with the sequential nature of handwriting scripts, attention-based models, and a mixture of both there still is a long way to go. During the processing of the input text image, recurrent sequential models take into account past information, as Long-Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), past and future information, as Bidirectional LSTM (BLSTM) networks, and multi-directional information, as Multi-Directional LSTM (MDLSTM) networks. These models have improved thanks to the inclusion of convolutional layers, which proved to be able to extract informative features from data (Puigcerver, 2017) (Retsinas et al., 2022). Sequential model capabilities have been enhanced by using attention mechanisms that weigh more the “most important” input data (Bluche et al., 2017). Gating mechanisms borrowed from LSTMs and GRUs were also used to-

^a <https://orcid.org/0000-0001-9937-7112>

^b <https://orcid.org/0000-0001-8110-1791>

^c <https://orcid.org/0000-0002-4508-1540>

^d <https://orcid.org/0000-0001-8992-9243>

*Equal contributions, authors listed in alphabetical order

gether with Convolutional Neural Networks (CNNs) to retain relevant information, showing competitive results compared to more classical Convolutional Recurrent Neural Network (CRNN) models (Coquenot et al., 2019). However, non-sequential models with attention mechanisms have outperformed sequential ones (Li et al., 2021).

Relaxation labeling procedures are parallel iterative processes able to simultaneously take data into account. Data is therefore processed in a non-sequential manner. This feature enables the model to learn the context in relation to all characters in the text line. In addition, they are able to refine the outputs concurrently, and not just one at a time. These characteristics were introduced with attention-based non-sequential models based on the (Vaswani et al., 2017) Transformer architecture.

Although the models mentioned above are efficient at solving the problem of sequence labeling (as in the case of handwriting recognition), how to properly embed and use contextual information is still an open question. Relaxation labeling processes proved to be able to include useful contextual information, as they reduce both local ambiguities and achieve global consistency. These are in fact parallel iterative labeling processes capable of performing the exploitation of contextual information.

Relaxation labeling processes for consistent labeling represented a fundamental model in the past in solving tasks such as text recognition (Goshtasby and Ehrich, 1988). Research, however, shifted its focus to neural network-based approaches (Lombardi and Marinai, 2020). This was probably due to one of the main disadvantages of these processes: the need to define the compatibility coefficients a-priori. To address this problem, (Pelillo and Refice, 1994) showed that a forward propagation error strategy is effective in learning good compatibilities in the field of Natural Language Processing.

In this paper, we include relaxation labeling processes in a well-established neural network architecture in the HTR field, the CRNN, to further refine the labeling output. We use a novel learning scheme for the parameters, i.e. the compatibility coefficients, of relaxation labeling processes to be learned together with those of the neural network. We conducted experiments which demonstrated the ability of relaxation labeling processes to improve the model generalisation capability.

2 HANDWRITING RECOGNITION MODEL

In this Section, we revise the theoretical framework of relaxation labeling and describe our proposed solution for handwritten text recognition.

2.1 Consistent Labeling Problem

Many models have been introduced to solve the so-called *consistent labeling problem*, a class of problems widely studied in computer vision, pattern recognition and artificial intelligence fields (Haralick and Shapiro, 1979). These problems involve labeling a set of objects such that certain domain-specific constraints must be satisfied: given a set of objects, there exist labeling configurations that are impossible. For example, in the case of character-level labeling in a line of English text, if we were to decide to label with a character knowing that the preceding character is a comma, we would choose a space. Indeed, the character space always follows such punctuation. Attempts in formalizing the notion of consistent labeling culminated in the seminal paper of Hummel and Zucker (Hummel and Zucker, 1983), who developed a formal theory of consistency that later turned out to have intimate connections with non-cooperative game-theory (Miller and Zucker, 1991). The theory generalizes the classical constraint satisfaction problem, where the constraints are boolean, to soft compatibility measures and probabilistic labeling assignments (Rosenfeld et al., 1976). It is possible to solve the first kind of problem through *discrete* relaxation labeling processes instead *continuous* relaxation labeling, also named *probabilistic* relaxation labeling processes, can solve the second one. We will use this second name in the case of soft compatibility measures.

In this paper, the second case scenario is considered, as it is better suited for unsegmented data. Label pairs cannot be fully compatible or incompatible. Compatibility coefficients must not be logical assertions but weighted values representing relative preferences.

To consistently label our data, the classical *relaxation labeling procedures* of (Rosenfeld et al., 1976) are used, as they present funded theoretical properties (Pelillo, 1997): when a specific condition of symmetry is met, such processes possess a Lyapunov function which drives them towards the nearest consistent solution. Furthermore, even if the symmetry condition is relaxed, more of the essential dynamical properties of the algorithm continues to hold.

2.1.1 Relaxation Labeling Processes

Here we introduce some necessary information about the relaxation labeling algorithms of (Rosenfeld et al., 1976).

As stated in the previous Section, relaxation labeling processes are iterative procedures that can take advantage of contextual information to label in a consistent manner a set of objects. The contextual information is embedded in the so-called *compatibility coefficients*. The compatibility coefficients are the non-negative real-valued elements in the matrix of compatibility coefficients, which we name R . Given $B = \{b_1, \dots, b_n\}$ the set of objects to label and $\Lambda = \{1, \dots, m\}$ the set of possible labels, R is defined in terms of a $n \times n$ block matrix:

$$R = \begin{bmatrix} R_{11} & \dots & R_{1n} \\ \vdots & \ddots & \vdots \\ R_{n1} & \dots & R_{nn} \end{bmatrix}, \quad (1)$$

where each R_{ij} , with i and $j = 1, \dots, n$, is a $m \times m$ matrix:

$$R_{ij} = \begin{bmatrix} r_{ij}(1,1) & \dots & r_{ij}(1,m) \\ \vdots & \ddots & \vdots \\ r_{ij}(m,1) & \dots & r_{ij}(m,m) \end{bmatrix}. \quad (2)$$

Each coefficient $r_{ij}(\lambda, \mu)$ measures the strength of compatibility between λ assigned to object b_i and μ to object b_j . High values correspond to compatibility, and low values correspond to incompatibility.

Let's assume $p_i(\lambda)$ to be the probability of an object i to be labeled with λ . The relaxation labeling processes are defined by the two formulas:

$$q_i^{(\tau)}(\lambda) = \sum_j \sum_{\mu} r_{ij}(\lambda, \mu) p_j^{(\tau)}(\mu), \quad (3)$$

$$p_i^{(\tau+1)}(\lambda) = \frac{p_i^{(\tau)}(\lambda) q_i^{(\tau)}(\lambda)}{\sum_{\mu} p_i^{(\tau)}(\mu) q_i^{(\tau)}(\mu)}, \quad (4)$$

with τ the iteration step of the processes. (Formula 3) is the *relaxation operator* that considers the context coded into the compatibility coefficients in R . $q_i^{(\tau)}(\lambda)$ represents the support of the context to labeling object b_i with label λ .

Iterating (Formulas 3-4), we would obtain a refined probability distribution for the objects, resulting in a less ambiguous labeling output.

The iterative procedure can be stopped after the updates become small in the norm or after a predefined number of steps.

2.1.2 Proposed Model and Learning Scheme

The proposed model is depicted in Figure 1. The architecture is composed of a neural backbone (baseline), taken from (Retsinas et al., 2022), and the relaxation labeling module. The baseline model comprises two main modules and an interface one. The first one is in charge of extracting visual features from images. The second learns the sequential patterns from data. These modules are connected through a flattening one to obtain a 1D-sequence of feature vectors from 2D-sequence feature representations because the recurrent head needs 1D-sequence data as input.

The relaxation labeling module is applied before the recurring one to avoid running into missing contextual information that can be present when using recurring layers.

To learn the parameters, we use the backward propagation through time, which is guaranteed to produce equivalent results to the forward propagation scheme developed in (Pelillo and Refice, 1994) but has the advantage of being computationally efficient (Baydin et al., 2018). This learning scheme is commonly used to train neural network models (Guo, 2013), (Werbos, 1990).

In Figure 2, we report the computational graph of the relaxation labeling module after defining the number of iterations \mathcal{T} . In this way, it is possible to see the architecture in the unfolded representation.

We modify the loss of (Retsinas et al., 2022) with the term involving the relaxation labeling procedures to train the architecture. All the used losses are CTC losses to satisfy the unsegmented data processing. The loss of (Retsinas et al., 2022) comprises two terms, \mathcal{L}_{End} which is the loss at the end of the architecture, and $\mathcal{L}_{Shortcut}$ which is the loss applied after the feature extractor module, the flattening one and what the authors called 'CTC Shortcut', a 1D-convolution. The 'CTC Shortcut', with $\mathcal{L}_{Shortcut}$, has been introduced to assist in training the network. The new term that we introduce to the loss, \mathcal{L}_{Relax} , allows learning the relaxation labeling compatibility coefficients together with those of the entire convolutional part of the neural network. The loss is in fact given by the following formula:

$$\begin{aligned} \mathcal{L}(\rho_{conv}, \rho_{rec}, R; s) = & \mathcal{L}_{End}(\rho_{conv}, \rho_{rec}; s) + \\ & \alpha \mathcal{L}_{Shortcut}(\rho_{conv}; s) + \\ & \beta \mathcal{L}_{Relax}(\rho_{conv}, R; s), \end{aligned} \quad (5)$$

with ρ_{conv} , ρ_{rec} representing a generic parameter of the convolutional and recurrent parts of the architecture, respectively. R is the compatibility matrix, and s is the transcription of the input line. Finally, α and β are hyperparameters, weighting the CTC loss of

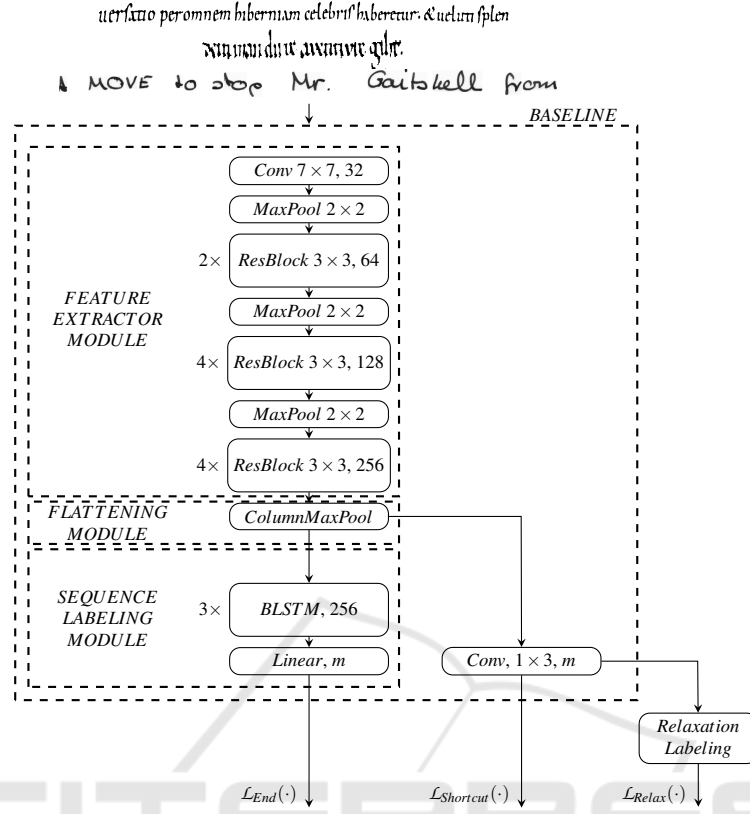


Figure 1: The architecture of our model. *ColumnMaxPool* represents the max pooling applied vertically to flatten the extracted feature maps, and m is the number of labels.

Table 1: Cardinality for the training, validation and test sets of the datasets.

Dataset	Train. Set	Valid. Set	Test Set
Saint Gall	468	235	707
Parzival	2,237	912	1,328
IAM	6,482	976	2,915

the CTC Shortcut and the relaxation labeling module, correspondingly.

3 EXPERIMENTS

In this Section we briefly introduce the datasets used and the pre-processing and data augmentation techniques performed. We, then, explicitly state the settings used for the experiments. Finally, a detailed explanation of the baseline architecture, the relaxation labeling module and how these are combined is reported.

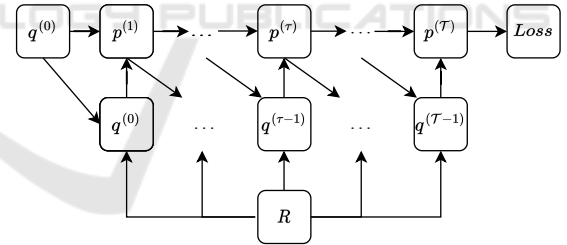


Figure 2: Computational graph of the unfolded relaxation labeling module.

3.1 Datasets

We use both historical and modern datasets with line-level transcription: the Saint Gall (Fischer et al., 2011), the Parzival (Fischer et al., 2012) and the IAM dataset (Marti and Bunke, 2002). Datasets are available from the Research Group on Computer Vision and Artificial Intelligence at the University of Bern¹. The Saint Gall dataset comprises images of manuscripts of the 9th century written in Latin, the Parzival dataset contains manuscript images of the 13th century written in German, the IAM dataset con-

¹<https://fki.tic.heia-fr.ch/databases>

Table 2: Experiments on the Saint Gall Dataset. Validation and test CER and WER are reported.

Model	Val. CER	Test CER	Val. WER	Test WER
Baseline	4,56%	4,76%	32,70%	33,95%
B. with Relax	4,43%	4,65%	31,95%	33,29%

Table 3: Experiments on the Parzival Dataset. Validation and test CER and WER are reported.

Model	Val. CER	Test CER	Val. WER	Test WER
Baseline	1,29%	1,46%	5,60%	6,46%
B. with Relax	1,19%	1,37%	5,15%	6,05%

Table 4: Experiments on the IAM Dataset. Validation and test CER and WER are reported.

Model	Val. CER	Test CER	Val. WER	Test WER
Baseline	4,00%	5,57%	14,36%	18,57%
B. with Relax	3,77%	5,50%	13,85%	18,50%

Table 5: Examples of refinement using the relaxation labeling module for all the datasets. Grey-filled bounding boxes highlight errors.

Sample Image (csg562-042-24)	
GT	Vocavit deinde unum e fribus& eum interrogavit quid
Without Relaxation Labeling	Vocavit deinde unum efribus & eum interrogavit quid
With Relaxation Labeling	Vocavit deinde unum e fribus& eum interrogavit quid
Sample Image (d-008b-025)	
GT	Gahm8reth der werde man.
Without Relaxation Labeling	Gahm8reth der werdet aan .
With Relaxation Labeling	Gahm8reth der werde man.
Sample Image (d04-012-10)	
GT	word of God .
Without Relaxation Labeling	word of hod
With Relaxation Labeling	word of God .

tains forms of handwritten modern English text from 657 different writers. The datasets have a different alphabet cardinality: 50, 95, and 79, respectively. Table 1 reports the cardinality of the partitions of the datasets. Transcription errors are present in all the datasets, a problem that is affecting the performance of the models (Aradillas et al., 2020).

3.2 Data Augmentation

We apply the best practices defined in (Retsinas et al., 2022) as image centring, left and right padding with the median intensity. We perform classical data augmentation techniques: affine transformations as rotations and translations, and gaussian blur filter of kernel 3×3 with a randomly chosen standard deviation

in the range $\sigma \in [1.0, 2.0]$. The images are resized to 128×1024 ($H \times W$).

3.3 Settings

The same setting of (Retsinas et al., 2022) is used to train the architecture: the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of $1E-3$ and weight decay of $5E-5$, a fixed number of epochs equal to 240 while decreasing the learning rate of 0.1 at specific epoch numbers (120 and 180) and $\alpha = 0.1$ weighting the CTC Shortcut contribution to the loss (cf. Equation (5)). To perform the analysis, we first train the entire network considering only the baseline architecture, then add the relaxation labeling module with different numbers of iterations. The con-

sidered numbers of iterations are equal to $1, \dots, 5, 10$ and 15 . For each architecture realisation, we perform hyperparameter optimization varying β in the set $\{0.1, 0.3, 0.5, 1.\}$. We also set a maximum wall time for the training of 24 hours. A batch size of 20 is used with 4 NVidia Tesla V100 16 Gb GPUs.

3.4 Baseline Model

As already mentioned, the baseline model we used is taken from (Retsinas et al., 2022). It is made of a CRNN. Specifically, the first convolutional layer has a kernel size 7×7 , and presents 32 filters. It is followed by a sequence of 3×3 ResNet blocks: 2 ResNet blocks with 64 filters, 4 ResNet blocks with 128 filters, and 4 ResNet blocks with 256 filters. Both the standard convolution and ResNet blocks are followed by ReLU activations, Batch Normalization and dropout. Between the sequences of blocks 2×2 maxpooling layer with stride 2 is set to reduce the feature map dimension. Then, the 3D feature map is reported to a sequence thanks to a maxpool by column. The recurrent head takes in input the aforementioned sequence. It consists of three stacked BLSTM layers, with a hidden size of 256. Finally, a Fully Connected (FC) layer is used to report the output dimension to the number of characters in the alphabet (in addition with the blank character, needed for the CTC loss (Graves et al., 2006)). The scores are then reported to a probability distribution thanks to the Softmax function (Bridle, 1989). To aid the network in the training process, a ‘CTC Shortcut’ was also used, consisting of a convolutional layer with kernel dimension 1×3 and a number of channels equal to the number of classes where another CTC loss is applied to help training.

3.5 Relaxation Labeling Module

The *probabilistic* relaxation labeling procedures are applied to the full-text line image to consider the whole line context. We apply it after the convolutional part of the architecture (see Figure 1) to include the context before the recurring part in order to avoid missing contextual information that can occur when training recurring layers. The final mapping of the convolutional part, including the 1D-convolution 1×3 , is of dimension $m \times 1 \times 128$ ($m \times H \times W$). The mapping elements represent the objects to label $B = \{b_1, \dots, b_n\}$, with $n = 128$. The labels m are all the characters in the respective dataset’s alphabet and the blank character necessary for the CTC loss.

The iteration number \mathcal{T} must be chosen beforehand. Models were released that performed optimally

even using only one iteration, $\mathcal{T} = 1$ (Pelillo and Refice, 1994). As already mentioned, the iteration number \mathcal{T} is treated like a hyperparameter to be tuned by doing various experiments.

3.6 Combined Architecture

The relaxation labeling module is placed at the end of the whole convolutional part of the baseline architecture (see Figure 1). Differently from (Pelillo and Refice, 1994), we do not make simplifying assumptions as the neighbourhood hypothesis (i.e. assuming that objects interact within a neighbourhood) and the stationarity hypothesis (i.e. the compatibility coefficients do not depend on the absolute positions of the objects, but on their relative distance) not to constrain the learning of the compatibility coefficients. This is to make all objects interact with each other, without imposing restrictions.

Given the size of the compatibility matrix, we decided to remove the relaxation labeling module in inference time, dealing with the CRNN, only. The relaxation labeling module proves to be able to drive the network towards a more consistent labeling output (see Section 4).

4 RESULTS AND DISCUSSION

4.1 Quantitative Analysis

Tables 2 - 4 report the best results for the considered datasets. The relaxation labeling processes showed the ability to drive the network to a more consistent labeling, obtaining a lower validation and test CER and WER than the baseline model. We found the best choice for $\beta = 0.1$, for all the architectures and all the datasets. Instead, the number of unfolds for the relaxation labeling module differs for each dataset. In the case of the Saint Gall and the IAM dataset, the best number of relaxation labeling processes iterations is $\mathcal{T} = 15$, and in the case of the Parzival dataset $\mathcal{T} = 3$.

4.2 Qualitative Analysis

Table 5 reports some cases where the relaxation labeling module increased the transcription accuracy for all the datasets. It is possible to notice that the model can perform all types of modifications to the text line: character substitutions, insertions and deletions. The cases of samples csg562-042-24, d-008b-025 and d04-012-10 report cases of substitutions. Insertions are present in the cases of csg562-042-24 and

d04-012-10. Finally, an example of deletion is present in the case of sample csg562-042-24. As already mentioned, errors are present in the datasets. In the IAM dataset, the GT transcriptions have many more spaces than the correct ones, especially before punctuation marks. Moreover, errors in the transcription are present, such as in the case of d04-012-10 where the GT transcription contains the final dot not present in the image itself.

5 CONCLUSIONS

We empirically demonstrated that relaxation labeling processes help in generalisation abilities for well-established architectures in the HTR field, the CRNNs. Such processes can drive the network towards a more consistent labeling output in all the datasets considered. They improve the results in terms of both validation and test CER and WER. As a future work we consider to compare the relaxation labeling module with attention mechanisms, which have a similar role of contextual processing from the context. Finally, we plan to conduct a more extensive comparison of our proposed method with other backbones, to consistently evaluate the performance improvement provided by relaxation labeling.

ACKNOWLEDGEMENTS

We would especially like to thank Gregory Sech for his valuable suggestions and his past work on Relaxation Labeling processes. Also, we want to thank the Italian Institute of Technology (IIT) for the possibility to use their High-Performance Computing (HPC) as it allowed for faster experimentation.

REFERENCES

- Aradillas, J. C., Murillo-Fuentes, J. J., and Olmos, P. M. (2020). Improving offline htr in small datasets by purging unreliable labels. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 25–30. IEEE.
- Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43.
- Bluche, T., Louradour, J., and Messina, R. (2017). Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1050–1055. IEEE.
- Bridle, J. (1989). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2.
- Coquenat, D., Soullard, Y., Chatelain, C., and Paquet, T. (2019). Have convolutions already made recurrence obsolete for unconstrained handwritten text recognition? In *2019 International conference on document analysis and recognition workshops (ICDARW)*, volume 5, pages 65–70. IEEE.
- Fischer, A., Frinken, V., Fornés, A., and Bunke, H. (2011). Transcription alignment of latin manuscripts using hidden markov models. In *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, pages 29–36.
- Fischer, A., Keller, A., Frinken, V., and Bunke, H. (2012). Lexicon-free handwritten word spotting using character hms. *Pattern recognition letters*, 33(7):934–942.
- Goshtasby, A. and Ehrich, R. W. (1988). Contextual word recognition using probabilistic relaxation labeling. *Pattern Recognition*, 21(5):455–462.
- Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.
- Guo, J. (2013). Backpropagation through time. *Unpubl. ms., Harbin Institute of Technology*, 40:1–6.
- Haralick, R. M. and Shapiro, L. G. (1979). The consistent labeling problem: Part i. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):173–184.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Hummel, R. A. and Zucker, S. W. (1983). On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3):267–287.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Li, M., Lv, T., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., and Wei, F. (2021). Trocr: Transformer-based optical character recognition with pre-trained models. *arXiv preprint arXiv:2109.10282*.
- Lombardi, F. and Marinai, S. (2020). Deep learning for historical document analysis and recognition—a survey. *Journal of Imaging*, 6(10):110.
- Marti, U.-V. and Bunke, H. (2002). The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46.
- Miller, D. A. and Zucker, S. W. (1991). Copositive-plus lemke algorithm solves polymatrix games. *Operations research letters*, 10(5):285–290.
- Pelillo, M. (1997). The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision*, 7(4):309–323.

- Pelillo, M. and Refice, M. (1994). Learning compatibility coefficients for relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):933–945.
- Puigcerver, J. (2017). Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 67–72. IEEE.
- Retsinas, G., Sfikas, G., Gatos, B., and Nikou, C. (2022). Best practices for a handwritten text recognition system. In *International Workshop on Document Analysis Systems*, pages 247–259. Springer.
- Rosenfeld, A., Hummel, R. A., and Zucker, S. W. (1976). Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, (6):420–433.
- Teslya, N. and Mohammed, S. (2022). Deep learning for handwriting text recognition: Existing approaches and challenges. In *2022 31st Conference of Open Innovations Association (FRUCT)*, pages 339–346. IEEE.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

