Published in *Document analysis and recognition ICDAR 2023 ; Proceedings of the 17th International Conference, 21-26 August 2023, San José, CA, USA* , which should be cited to refer to this work.

DOI:https://doi.org/10.1007/978-3-031-41676-7_23

The Bullinger Dataset: A Writer Adaptation Challenge

Anna Scius-Bertrand[1,2], Phillip Ströbel[3], Martin Volk[3], Tobias Hodel[4], and Andreas Fischer[1,2]

[1] iCoSys Institute, University of Applied Sciences and Arts Western Switzerland, Fribourg, Switzerland
[2] DIVA Group, University of Fribourg, Switzerland
{anna.scius-bertrand, andreas.fischer}@hefr.ch
[3] Department of Computational Linguistics, University of Zurich, Switzerland
{pstroebel, volk}@cl.uzh.ch
[4] Walter Benjamin Kolleg, University of Bern, Switzerland
tobias.hodel@unibe.ch

Abstract. One of the main challenges of automatically transcribing large collections of handwritten letters is to cope with the high variability of writing styles present in the collection. In particular, the writing styles of non-frequent writers, who have contributed only few letters, are often missing in the annotated learning samples used for training handwriting recognition systems. In this paper, we introduce the Bullinger dataset for writer adaptation, which is based on the Heinrich Bullinger letter collection from the 16th century, using a subset of 3,622 annotated letters (about 1.2 million words) from 306 writers. We provide baseline results for handwriting recognition with modern recognizers, before and after the application of standard techniques for supervised adaptation of frequent writers and self-supervised adaptation of non-frequent writers.

Keywords: Handwriting Recognition · Writer Adaptation · Historical Documents · Handwritten Letters

1 Introduction

Handwriting recognition remains a mostly unsolved problem and a very active field of research, because it challenges pattern recognition and machine learning techniques in various ways: Even when considering samples written in the same language and time period, there is a high variability in character shapes and character connections to model, especially in the case of cursive handwriting. When changing the language, there is a distribution shift regarding language models, even when the same set of characters are used. For historical documents [4], additional difficulties include the absence of timing information, which is only available for modern on-line handwriting with an electronic pen, degraded paper or parchment due to old age, which leads to artifacts on the scanned page images, and a large number of different languages, scripts, and time periods to consider.

Therefore, handwritten text recognition (HTR) usually targets a very specific type of handwriting, e.g. a historical manuscript written by only a few different hands, with similar imaging conditions across the scans, the same language, etc. and is trained with a large amount of annotated learning samples from the same type of handwriting.

In this paper, we introduce a novel challenge for HTR in the context of a comprehensive digitization project [2] in Switzerland that aims to create a digital edition of a large collection of historical letters, namely the Bullinger letters, which include about 12,000 letters written or received by Heinrich Bullinger (1504-1575), an important Swiss Reformer. He was in contact with over 1,000 persons, which introduces a high variability of writing styles, as well as differences in writing support and writing instruments. Furthermore, the letters are not only written in Latin but also in a premodern form of German, and sometimes the two languages are mixed, while the language might change either from paragraph to paragraph or even mid-sentence (i.e., code-switching). Over the past years, transcription and transcription alignment efforts have been focused on the most frequent writers, i.e. Bullinger himself and persons who have written a considerable number of letters to him. However, there are thousands of letters from non-frequent writers, whose writing styles are not present in the annotated training material. Therefore, one of the most intriguing problems is that of writer adaptation: *"Is it possible to adapt a generic HTR system to the specific writing style of a non-frequent writer, who is not represented in the training data, such that the HTR performance is improved?"* The same question, although less challenging, can also be asked for frequent writers, who are represented in the training set and may also profit from an adaptation to their particular style of writing.

1.1 Related Work

There is a rich body of literature on the topic of writer adaptation for HTR. To name just a few, early examples include [17], where the adaptation is performed based on writer-specific allographs that are used to re-evaluate the output of an HTR system, and [3], where unsupervised clustering is used to estimate Gaussian mixture models that are specific to a writing style. In [6], a self-training approach is pursued to improve the performance of an HTR system by adapting it to the recognition output of unlabeled samples. The work presented in [7] employs a keyword spotting strategy to adapt an HTR system trained for modern handwriting to historical handwriting. More recent attempts to perform transfer learning are reported in [8,10]. A competition organized on the READ dataset specifically included the problem of writer adaptation with respect to 22 different hands, 5 of which are used both in the training and the test set to investigate supervised adaptation [21]. The best results are obtained when adapting both the optical and the language models, and when including data augmentation [19]. Targeting the more difficult case of unsupervised adaptation, a style adaptation at multiple abstraction layers of a deep convolutional model is proposed in [22]. Another recent unsupervised adaptation scheme is based on fully synthetic training data [11].

1.2 Contribution

In the present work, we do not introduce a novel method for writer adaptation. Instead, we introduce the Bullinger dataset for writer adaptation [1] and establish baseline results using state-of-the-art HTR systems with standard adaptation strategies, i.e. fine-tuning a generic HTR system on the training data of the frequent writers, and fine-tuning the models on confidently transcribed text lines of non-frequent writers, following a self-training methodology [6]. The dataset is publicly and freely available for developing and comparing novel approaches to writer adaptation.

When comparing the Bullinger dataset with other datasets used for writer adaptation research, we can highlight the difficulty of the handwriting itself (cf. Figure 1), which is also difficult to read for human experts, and the large number of over one million words, which is suitable for experiments with deep learning models from the current state of the art. Table 1 provides a comparison with other related research datasets. Note that only about a quarter of all letters are currently included in the Bullinger dataset, representing the progress of the digitalization project. The total number of writers is over 1,000 for the entire letter collection.

Dataset	**Number of words**	**Number of writers**
Georges Washinghton [13]	4,860	2
Parzival [5]	23,478	3
Rimes [9]	66,978	1,300
READ [21]	98,239	22
CVL [12]	99,902	310
IAM Handwriting [15]	115,320	657
Bullinger	1,241,714	306

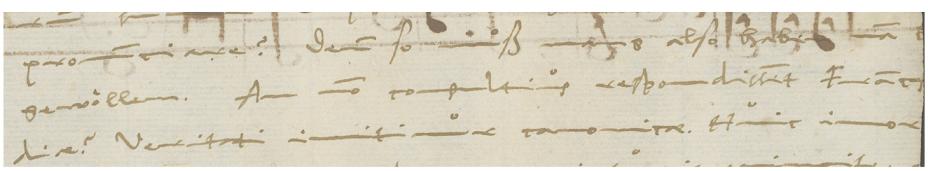
Table 1. Related research datasets for writer adaptation research.

In the remainder, we describe the handwriting present in the Bullinger letters and how it has been transcribed so far, introduce the HTR systems and writer adaptation techniques considered for the experiments, present the baseline results, and draw some conclusions.

2 Dataset

The Bullinger Digital project [2] aims to bring together all available resources about the comprehensive letter correspondence of Heinrich Bullinger (1504-1575), a Swiss Reformer, in a single database. For this purpose, all available meta-information, e.g. about the writers of the letters, but also scanned page images and existing transcriptions are brought together. Furthermore, the goal is to automatically align existing transcriptions with the page images and to use HTR to perform an automatic transcription for the remaining letters. Heinrich Bullinger was a key actor during the Reformation in Switzerland and Europe. His letter correspondence includes about 2,000 letters written by himself and about 10,000 letters that he has received from over 1,000 persons.

Certain writers, including Bullinger, exhibit writing styles that are very difficult to read, even for human experts. Figure 1 provides an example of Bullinger's handwriting. We can observe a mix of Latin and a premodern form of German phrases, abbreviations, and words that are very difficult to decipher without intimate knowledge of the handwriting, or access to a transcription. At the beginning of the third line, we can also observe a missing word in the transcription. It is due to an error of the automatic transcription alignment, which was performed using the Text2Image module of the Transkribus platform [16]. In general, the quality of the alignment is high, and thus the quality of the ground truth for handwriting recognition, but especially at the beginning and at the end of the text lines errors may arise due to word breaks. Furthermore, the transcription is not necessarily character-accurate, e.g. abbreviations are often written out in full. This noise in the automatically generated ground truth is an additional difficulty for training HTR systems.



pronunciaret denn sparnuß michs also habe man
gewo̊llen. An non consultius respondissent Franc
veritati invitimur canonicae. Hinc immor

Fig. 1. Text lines written by Bullinger with automatically aligned transcriptions.

Figure 2 illustrates the high variability of writing styles present in the Bullinger dataset. The first line shows three examples of how Bullinger writes his own name. They exhibit a considerable intra-writer variability. The remaining words are written by other persons, demonstrating changes in the writing style, writing support, and writing instrument.

Fig. 2. Different writing styles and different forms of the word "Bullinger". The writer IDs are indicated in the bottom-left corner of each automatically segmented word. All words of the first line are written by the same writer, Bullinger himself.

Note that the sample word images were automatically cut out from text lines that have been processed by an HTR system and may contain segmentation errors. The text lines themselves were cut out from the scanned page images according to polygonal boundaries provided by Transkribus' layout analysis system. The special background pattern around the text lines is added artificially instead of white background, to make the background more homogeneous for HTR.

3 Methods

3.1 Handwriting Recognition

We consider two state-of-the-art models for handwriting recognition, namely PyLaia [18] and HTR-Flor [20]. They both consider deep convolutional layers to extract features from text line images, followed by bidirectional recurrent layers with connectionist temporal classification (CTC) loss to analyze the features from left-to-right as well as right-to-left to recognize character sequences. They differ in the composition of the layers as illustrated in Figure 3. To reduce the number of trainable parameters, HTR-Flor uses gated convolution and bidirectional gated recurrent units (BGRU) instead of standard convolution and bidirectional long short-term memory cells (BLSTM). In effect, HTR-Flor only has around 820 thousand parameters, which is significantly less than the 9.6 million parameters of PyLaia. Nevertheless, both models achieve similar results on several benchmark datasets [20].

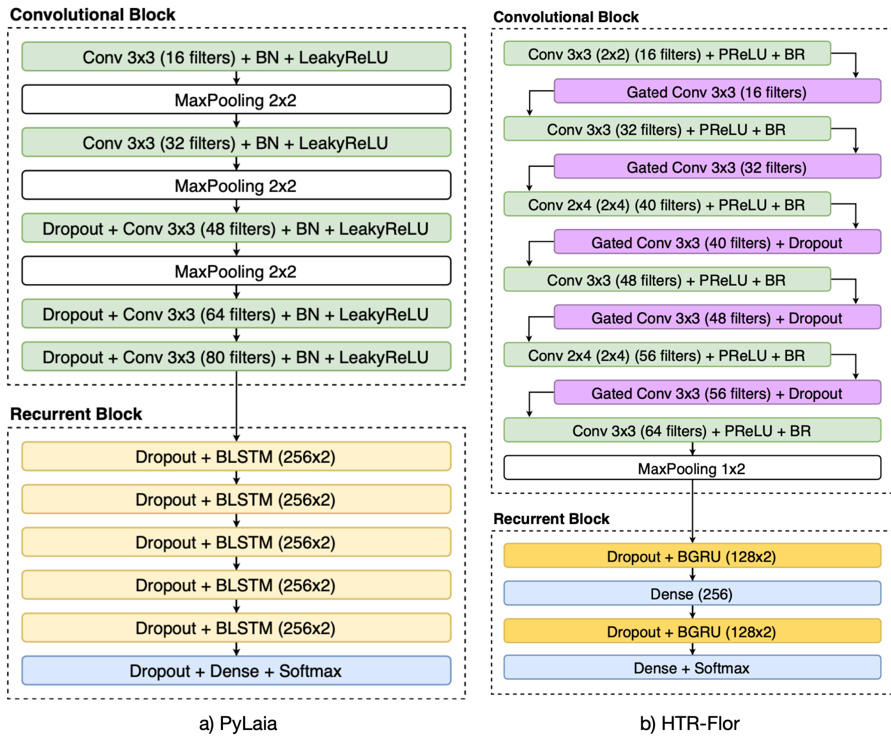


Fig. 3. Architectures of the two HTR systems used: PyLaia [18] and HTR-Flor [20]. Both figures are taken from [20].

3.2 Writer Adaptation

We consider two standard writer adaptation methods to study their impact on the HTR performance. The first method is used for frequent writers, for which some of the letters were transcribed and are part of the training set. In this case, we train a generic HTR system on all letters and then fine-tune it on the training letters of the frequent writer in order to adapt the model to the specific writing style.

The second method is used for non-frequent writers, for which none of the letters have been transcribed and therefore no training material is available. In this case, we follow the self-training approach proposed in [6]. We start again with a generic HTR system trained on all letters and apply it to the letters of the non-frequent writer. Afterwards, we compute a confidence measure $C(s)$ for the predicted characters sequences $s = c_0, \ldots, c_N$,

$$C(s) = \prod_{i=0}^{N} p(c_i) \ , \tag{1}$$

where $p(c_i)$ is the softmax probability of the character c_i according to the CTC decoding. Afterwards, we sort the character sequences of all text lines according to their confidence and use the most confident P percent of the text lines as a new training set for fine-tuning the generic HTR system.

4 Experimental Evaluation

4.1 Database Setup

To study the impact of writer adaptation, we consider text line images from a subset of 3,622 letters by 306 writers with automatically aligned transcriptions, which are used as ground truth for the HTR experiments.

As illustrated in Figure 4, the database is split as follows for the Bullinger writer adaptation challenge: First, we sort the writers according to their number of letters, observing a Zipf distribution with only few frequent writers and a large number of non-frequent writers. Then, using a threshold of 5 letters, we distinguish two groups of writers:

- **Frequent writers:** Writers with at least 5 letters.
- **Non-frequent writers:** Writers with less than 5 letters.

There are 106 frequent writers in total. We use the first 80% of their letters for training, the next 10% for validation (optimization of hyper-parameters), and the final 10% for testing. For the non-frequent writers, we select the next 200 writers in the sorted list of writers to compose a second test set of similar size. In this experimental setup, the test set for frequent writers estimates how well HTR performs for known writers, where several of their letters have been transcribed for training, and the test set for non-frequent writers estimates how well HTR performs for unknown writers, whose writing styles are not present during training. This scenario reflects the real situation in the Bullinger Digital project [2], where the transcription efforts are directed towards the most important (most frequent) writers. Table 2 shows the exact repartition of writers, letters, pages, text lines, and words across the different sets. The training set has a considerable size of 109,627 text lines with 876,003 words. After removing some very rare characters, we retain a total of 78 distinct characters in the database including the space character.

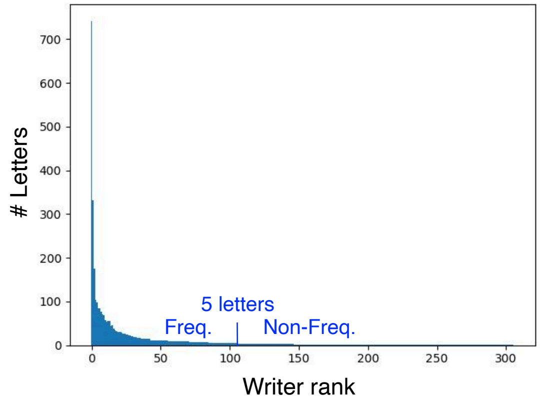


Fig. 4. Database setup. The graph shows the number of letters per writer. Frequent writers (Freq.) have five or more letters and non-frequent writers (Non-Freq.) have less than five letters.

	Training	**Validation**	**Test Freq.**	**Test Non-Freq.**	**Total**
# of writers	106	106	106	200	306
# of letters	2,581	337	337	367	3,622
# of pages	5,927	806	787	873	8,393
# of lines	109,627	14,516	15,368	15,735	155,246
# of words	876,003	122,211	115,289	128,211	1,241,714

Table 2. Database setup: Distribution of writers, letters, pages, text lines and words for frequent writers (Freq.) and non-frequent writers (Non-Freq.).

4.2 HTR Setup

The hyper-parameters of the HTR systems were optimized on the validation set during preliminary experiments. They have been fixed to the same values for HTR-Flor and PyLaia. The text line images are resized to a height of 128 pixels, keeping the aspect ratio, and in addition to the three RGB channels we add a fourth channel with a binary version of the image obtained by means of a global Otsu threshold. We consider 256 hidden units for the LSTMs/GRUs, a dropout of 0.3 in the recurrent layers, and a mini-batch size of 64. The learning rate is optimized with AdamW [14] using a weight decay of 0.0001, $\beta_1 = 0.9$, and $\beta_2 = 0.98$. The peak learning rate is 0.00055. The HTR systems are trained for 100 epochs until convergence and the best model epoch is chosen with respect to the character error rate on the validation set. We do not observe a significant overfitting effect. Training one epoch with two NVIDIA TITAN RTX cards took around 23 minutes for HTR-Flor and around 30 minutes for PyLaia.

For the CTC, we use 78 character tokens, decode greedily by considering the character with maximum probability at each time step, and remove repeated consecutive characters. Note that we do not use a lexicon for text recognition, because of the presence of both Latin and German texts, spelling variants, and abbreviations. Instead, the text is transcribed character by character. In the present baseline experiments, we only focus on the optical model and the implicit language model learned by the recurrent layers on the training set. No explicit character language model is used.

4.3 Writer Adaptation Setup

For writer adaptation, we first train a generic system on the entire training set. Afterwards, the system is adapted as follows:

– **Frequent writers:** The training letters of the writer are used to further fine-tune the generic system. Either the training is **continued** with the same, small learning rate or the training is **restarted** with the initial, high learning rate. Either 10 or 20 epochs of training are pursued and the best number of epochs is determined on the validation letters of the writer.
– **Non-frequent writers:** Self-training is performed by recognizing the test letters of the writer with the generic system. Afterwards, the automatic transcriptions are sorted by recognition confidence (see Section 3.2) and the top 50%, 75%, or 100% of the text lines are used as learning samples to further fine-tune the generic system. Training is continued for either 10 or 20 epochs (since the non-frequent writers have no annotated validation letters, it is not possible to determine the best epoch prior to 10 or 20).

4.4 Evaluation Measures

We use the standard measures of character error rate (CER) and word error rate (WER) to evaluate the HTR performance. They are calculated by computing the string edit distance between the recognition output and the ground truth, to obtain the number of substitution, deletion, and insertion errors. By dividing the number of character errors with the number of characters in the ground truth, we obtain the CER, and similarly the WER.

For measuring the impact of writer adaptation, we report absolute improvements, e.g. $CER_g - CER_a$, as well as relative improvements in percentage, e.g.

$$100 \cdot \frac{CER_g - CER_a}{CER_g} \ , \tag{2}$$

with CER_g the error rate of the generic system and CER_a the (typically lower) error rate of the adapted system.

	Frequent writers		**Non-Frequent writers**	
	HTR-Flor	**PyLaia**	**HTR-Flor**	**PyLaia**
CER	9.56	**8.36**	10.67	**9.85**
WER	33.72	**29.64**	37.56	**34.39**

Table 3. HTR performance. The best results are highlighted in bold.

	Frequent Writers	
Configuration	**HTR-Flor**	**PyLaia**
C-10	9.08	**3.70**
C-20	**9.76**	3.39
R-10	4.30	-0.05
R-20	6.80	1.56

Table 4. Frequent writer adaptation. Relative improvement of the CER in percentage for continuing training during 10 or 20 epochs (C-10 and C-20) and for restarting training during 10 or 20 epochs (R-10 and R-20). The best results of each HTR system are highlighted in bold.

4.5 Results

HTR performance. Table 3 shows the CER of the generic (non-adapted) HTR systems for frequent and non-frequent writers, respectively. The best results are obtained with PyLaia, which achieves 8.36% CER for frequent writers and 9.85% CER for non-frequent writers. HTR-Flor performs about one percent CER worse, which may be due to the reduced number of model parameters when compared with PyLaia, taking into account the large size of the training set. For both systems, the error rate for non-frequent writers is significantly higher, which is expected because the writing styles of the non-frequent writers are not present in the training set. When comparing the overall HTR performance with the results for HTR-Flor on the IAM database reported in [20], namely 3.98% CER, the increased difficulty of the Bullinger database becomes evident.

Frequent writer adaptation. Table 4 shows the results of frequent writer adaptation for different fine-tuning strategies, in terms of relative improvements of the CER. For both HTR systems, restarting with a high learning rate is significantly worse than continuing the fine-tuning with a low learning rate. In the case of PyLaia, restarting 10 epochs even leads to an increase in the CER. The largest gain is observed for HTR-Flor, where the relative reduction of the CER is 9.76%.

Non-frequent writer adaptation. Table 5 shows the results of non-frequent writer adaptation for different self-training and fine-tuning strategies. The gain

Non-Frequent Writers		
Configuration	**HTR-Flor**	**PyLaia**
C-10, S-50%	2.65	0.44
C-10, S-75%	2.68	0.51
C-10, S-100%	2.01	**0.71**
C-20, S-50%	**2.88**	-1.84
C-20, S-75%	2.54	-2.99
C-20, S-100%	0.79	-4.42

Table 5. Non-frequent writer adaptation. Relative improvement of the CER in percentage for continuing training during 10 or 20 epochs (C-10 and C-20) and selecting the 50%, 75%, and 100% most confidently recognized text lines for self-training (S-50%, S-75%, and S-100%). The best results of each HTR system are highlighted in bold.

in performance is very limited for PyLaia, which clearly overfits to the self-labeled transcriptions when fine-tuning 20 epochs. The best results are achieved when selecting all self-labeled transcriptions and fine-tuning 10 epochs. HTR-Flor achieves the best result when selecting the 50% most confident text lines and fine-tuning 20 epochs. In this scenario, the relative improvement of the CER is 2.88%.

Detailed adaptation results. Table 6 provides a more detailed account for both frequent and non-frequent writer adaptation using the best fine-tuning and self-training strategies. Besides the improvements and relative improvements in CER and WER, we also indicate for how many writers the performance was improved. The improvements for frequent writers and the improvements of HTR-Flor for non-frequent writers are significant ($p < 0.05$). The improvements of PyLaia for non-frequent writers are not significant.

Overall, the adaptation results highlight a clear adaptation success for the frequent writers but only a limited success for the non-frequent writers. Even with the best fine-tuning and self-training configurations, the relative improvements in CER and WER remain very modest for the non-frequent writers.

5 Conclusion

The Bullinger dataset for writer adaptation introduced in this paper is a novel benchmark for developing and comparing writer adaptation methods. Its difficult handwriting, high variability in writing styles, and large size make it ideally suited for investigating writer adaptation with deep learning models from the current state of the art. The baseline results provided for the HTR-Flor and PyLaia architectures achieve up to 9.76% relative improvement of the CER for

	Frequent writers		**Non-Frequent writers**	
	HTR-Flor	**PyLaia**	**HTR-Flor**	**PyLaia**
Configuration	C-20	C-10	C-20, S-50%	C-10, S-100%
CER before	9.56	**8.36**	10.67	9.85
CER after	8.62	**8.05**	10.36	9.78
Improvement	**0.93**	0.31	0.31	0.07
Relative improvement	**9.76**	3.70	2.88	0.71
# writers improved	**97/106**	84/106	133/200	93/200
% writers improved	**91.51**	79.25	66.50	46.50
WER before	33.72	**29.64**	37.56	34.39
WER after	30.96	**28.72**	36.75	33.99
Improvement	**2.75**	0.93	0.81	0.39
Relative improvement	**8.17**	3.12	2.16	1.15
# writers improved	**96/106**	82/106	129/200	95/200
% writers improved	**90.57**	77.36	64.50	47.50

Table 6. Detailed adaptation results for the optimal system configurations. CER, WER, and improvements in percentage. The improvements for frequent writers and the improvements of HTR-Flor for non-frequent writers are significant ($p < 0.05$). The improvements of PyLaia for non-frequent writers are not significant.

supervised adaptation of frequent writers, but only up to 2.88% relative improvement of the CER for self-supervised adaptation of non-frequent writers.

Promising lines of research to improve over these baseline results include a conjoint adaptation of optical models and explicit language models, writing style clustering, data augmentation, and synthetic data generation, to name just a few.

The Bullinger project is still ongoing and we expect to be able to release more versions of the challenge in the future, increasing the size of the database and improving the quality of the ground truth.

Acknowledgements

This work has been supported by the Hasler Foundation, Switzerland. We would also like to thank Anna Janka, Raphael Müller, Peter Rechsteiner, Dr. Patricia Scheurer, David Selim Schoch, Dr. Raphael Schwitter, Christian Sieber, Martin Spoto, Jonas Widmer, and Dr. Beat Wolf for their contributions to the dataset and ground truth creation.

References

1. Bullinger Dataset for Writer Adaptation. Accessed on: 30.4.2023, `https://tc11.cvc.uab.es/datasets/BullingerDB_1`
2. Bullinger Digital project. Accessed on: 30.4.2023, `https://www.bullinger-digital.ch`
3. Fink, G.A., Plötz, T.: Unsupervised estimation of writing style models for improved unconstrained off-line handwriting recognition. In: 10th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR). pp. 1–6 (2006)
4. Fischer, A., Liwicki, M., Ingold, R. (eds.): Handwritten Historical Document Analysis, Recognition, and Retrieval – State of the Art and Future Trends. World Scientific (2020)
5. Fischer, A., Keller, A., Frinken, V., Bunke, H.: Lexicon-free handwritten word spotting using character hmms. Pattern recognition letters **33**(7), 934–942 (2012)
6. Frinken, V., Bunke, H.: Evaluating retraining rules for semi-supervised learning in neural network based cursive word recognition. In: Proc. 10th Int. Conf. on Document Analysis and Recognition (ICDAR). pp. 31–35 (2009)
7. Frinken, V., Fischer, A., Bunke, H., Manmatha, R.: Adapting BLSTM neural network based keyword spotting trained on modern data to historical documents. In: Proc. 12th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR). pp. 352–357 (2010)
8. Granet, A., Morin, E., Mouchère, H., Quiniou, S., Viard-Gaudin, C.: Transfer learning for handwriting recognition on historical documents. In: Proc. 7th Int. Conf. on Pattern Recognition Applications and Methods (ICPRAM). pp. 1–8 (2018)
9. Grosicki[1], E., Carre, M., Brodin, J.M., Geoffrois[1], E.: Rimes evaluation campaign for handwritten mail processing. In: 11th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR). pp. 1–6 (2008)
10. Jaramillo, J.C.A., Murillo-Fuentes, J.J., Olmos, P.M.: Boosting handwriting text recognition in small databases with transfer learning. In: Proc. 16th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR). pp. 429–434 (2018)
11. Kang, L., Rusinol, M., Fornes, A., Riba, P., Villegas, M.: Unsupervised writer adaptation for synthetic-to-real handwritten word recognition. In: Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 3502–3511 (2020)
12. Kleber, F., Fiel, S., Diem, M., Sablatnig, R.: CVL-database: An off-line database for writer retrieval, writer identification and word spotting. In: Proc. 12th Int. Conf. on Document Analysis and Recognition (ICDAR). pp. 560–564 (2013)
13. Lavrenko, V., Rath, T.M., Manmatha, R.: Holistic word recognition for handwritten historical documents. In: Proc. 1st Int. Workshop on Document Image Analysis for Libraries. pp. 278–287 (2004)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th Int. Conf. on Learning Representations (ICLR). pp. 1–18 (2019)
15. Marti, U.V., Bunke, H.: The iam-database: an english sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition **5**, 39–46 (2002)
16. Muehlberger, G., Seaward, L., Terras, M., et al.: Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study. Journal of Documentation **75**(5), 954–976 (2019)
17. Nosary, A., Heutte, L., Paquet, T.: Unsupervised writer adaptation applied to handwritten text recognition. Pattern Recognition **37**(2), 385–388 (2004)

18. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: Proc. 14th Int. Conf. on Document Analysis and Recognition (ICDAR). vol. 1, pp. 67–72 (2017)
19. Soullard, Y., Swaileh, W., Tranouez, P., Paquet, T., Chatelain, C.: Improving text recognition using optical and language model writer adaptation. In: Proc. 15th Int. Conf. on Document Analysis and Recognition (ICDAR). pp. 1175–1180 (2019)
20. de Sousa Neto, A.F., Bezerra, B.L.D., Toselli, A.H., Lima, E.B.: HTR-Flor: A deep learning system for offline handwritten text recognition. In: Proc. 33rd SIBGRAPI Conf. on Graphics, Patterns and Images (SIBGRAPI). pp. 54–61 (2020)
21. Strauß, T., Leifert, G., Labahn, R., Hodel, T., Mühlberger, G.: Icfhr2018 competition on automated text recognition on a read dataset. In: Proc. 16th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR). pp. 477–482 (2018)
22. Yang, H.M., Zhang, X.Y., Yin, F., Sun, J., Liu, C.L.: Deep transfer mapping for unsupervised writer adaptation. In: Proc. 16th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR). pp. 151–156 (2018)