A Comprehensive Comparison of Open-Source Libraries for Handwritten Text Recognition in Norwegian

Martin Maarand[1], Yngvil Beyer[2], Andre Kåsen[2], Knut T. Fosseide[3], and Christopher Kermorvant[1][0000−0002−7508−4080]

[1] TEKLIA, France
[2] National Library of Norway
[3] Lumex, Norway

Abstract. In this paper, we introduce an open database of historical handwritten documents fully annotated in Norwegian, the first of its kind, allowing the development of handwritten text recognition models (HTR) in Norwegian. In order to evaluate the performance of state-of-the-art HTR models on this new base, we conducted a systematic survey of open-source HTR libraries published between 2019 and 2021, identified ten libraries and selected four of them to train HTR models. We trained twelve models in different configurations and compared their performance on both random and scripter-based data splitting. The best recognition results were obtained by the PyLaia and Kaldi libraries which have different and complementary characteristics, suggesting that they should be combined to further improve the results.

Keywords: Handwriting recognition · Norwegian language · open-source.

1 Introduction

Thanks to the recent progress of handwritten text recognition (HTR) systems based on deep learning, the automatic transcription of handwritten documents has become a realistic objective for an increasing number of cultural heritage institutions. Archives and libraries are increasingly willing to include HTR systems in their digitization workflow similarly to their long standing use of OCR (optical character recognition) in the digitization of printed documents. Their goal is to index and make searchable all digitized documents, whether printed or handwritten. Several research projects such as READ/Transkribus [18] and eScriptorium [13] have shown that HTR can now be used at a large scale for automatically transcribing historical handwritten documents. They have demonstrated that high-level accuracy can be reached when HTR models are specifically trained on a representative sample of the target data that has been manually transcribed. However, if the goal is to automatically transcribe the complete collections of large libraries or archival institutions consisting of handwritten documents with a huge stylistic variety, manual annotation of a large representative sample rapidly

Fig. 1. A sample of pages from the dataset: letters from Henrik Ibsen (1872), Camilla Collett (1877) and Harriet Backer (1919).

becomes prohibitively costly. In such cases, generic recognition systems, independent from the writer, the period of time and even the language, are needed, as noted by [8].

However, cultural heritage institutions did not wait for the arrival of HTR engines to proceed with the transcription of their documents. Many collections have been researched and transcribed manually and sometimes even edited and published. These research or publishing projects are usually focused on one author, one historical period or one type of document, which means that the transcribed corpus is usually very homogeneous: all the documents are from the same author or from a limited number of authors or of the same type. These sets of transcribed documents can be used as a basis for the constitution of training corpora for HTR engines, but they contain an important bias: their selection was not carried out randomly. They are therefore not representative of all the documents in the collections of the institution, and the engines trained on these documents will consequently have a low generalization capacity. However, should we proceed with a new sampling and a new transcription of documents and put aside all these already transcribed documents?

Once the training corpus has been identified, the implementation of HTR processing requires the choice of a library and the training of the models. Today the technologies used by the state-of-the-art models are relatively homogeneous and based on the same Deep Learning algorithms. Several open-source libraries are available to train these models. The choice of one library over another is difficult for a non-expert because very few direct comparisons are published. Scientific articles generally present results for models obtained after an advanced expert optimization of parameters and hyper-parameters, the full details of which are not always available. The databases used for these comparisons are reference databases, prepared and normalized, which have been used for many years and on which the systems are over-optimized IAM[16], RIMES[2]. The complexities of the implementations to obtain satisfactory results are generally not evaluated.

We propose in this paper to study these different aspects using a new database of Norwegian handwritten documents. Our contributions are the following:

- we introduce and describe a new database of handwritten documents in Norwegian, manually transcribed and freely accessible;
- we present the result of a comprehensive survey of the libraries used in recent publications on handwriting recognition and a comparison of these libraries according to criteria allowing to guide the choice of one or several of them for a HTR project;
- we present a comparison of the error rates obtained on this new database by models trained with a selection of libraries;
- we study the generalization capabilities of these models to new writers on the Norwegian database.

2 Related work

Spoken by about five million people, mainly in Norway, Norwegian is considered a low-resource language. Even though some speech recognition or machine translation services are available online, the linguistic resources, spoken or written corpora, needed to develop automatic language processing tools are very limited. To the best of our knowledge, there are no digitized corpora of handwritten documents available in Norwegian [9]. An important work has been done within the Norwegian National Library to collect a corpus of electronic texts of sufficient size to train a BERT-type language model [14]. These language models can be used in other applications such as speech recognition in Norwegian [22]. The constitution of corpora allowing the training of named entity extraction models is also recent [10].

Regarding the comparison of handwriting recognition systems, the most common practice in the scientific literature is to compare a new system to the state-of-the-art on reference bases, reporting results published in other papers [24,3,12,28,11,4,5]. It is quite rare to see authors of new algorithms re-implementing and re-training state-of-the-art models for their evaluation [20,27], due to the significant work involved in re-implementing several models. This approach would not be very reliable for a comparison anyway, because authors tend to optimize their own system more than their competitors do, in order to maximize their chance to have their results published. Benchmarks of handwriting recognition services have been published for printed text recognition (OCR) [7], but only with generic models and services, without training. Another approach to promote the comparison of handwriting recognition systems is to organize competitions [25] or to publish datasets that can be used as benchmarks [26].

3 The Hugin-Munin dataset for HTR in Norwegian

3.1 Overview

Since the beginning of the millennium, the National Library of Norway (NLN) has been a heavy user of Optical Character Recognition (OCR) i.e. automatic recognition of characters in printed books. In the last couple of years, neural

networks have successfully improved recognition technology beyond printed text. This improvement has enabled the application of such technology to handwriting at a large scale. However, the heterogeneity of handwritten documents is a bigger challenge in handwritten text recognition (HTR) than in OCR. There are considerable variations between both writers and writing styles.

The digital instance of NLN is `nb.no/search`, a digital search engine. With this, users can discover and explore the wide range of texts that constitute the textual cultural heritage of Norway. Yet, for handwritten documents in the collection such a search function has not been available. One of the long-term goals of our present work is *searchability* in nb.no/search or more precisely a fulltext index. Searchability will streamline private archives and make this kind of material available in the same way as printed books. Another long-term goal is to provide readability or a reading support function, since it is not a given that users of today can read old handwriting.

The present work has profited greatly from earlier transcription efforts in different institutions in Norway. Both Collett and Kielland are part of an editorial philological series at the NLN[4], while for Ibsen and Munch the transcriptions have been generously shared with us by the University of Oslo[5] and the Munch Museum[6], respectively.

3.2 Dataset

The current dataset consists of private correspondences and diaries of 12 Norwegian writers with a significant representation in the collection at the NLN. All the documents were written between 1820 and 1950, and they are owned and digitized by the NLN. The various collected transcriptions have been converted to PAGE XML. The selected writers represent a variation in styles of handwriting and orthography.

Some of the writers were selected because they had already been transcribed in other projects, whereas others were selected due to on-going editorial philological projects or forthcoming dissemination activities, but also due to requests from the Norwegian research community[7].

The dataset consists of 164,922 words or tokens in 23,732 lines. It will be published open-source on Zenodo. The images will be distributed in a suitable format and the transcriptions in PAGE XML format.

We have defined two different splits for the experiments:

Random split : we randomly split all the pages of the dataset in 80% for training, 10% for validation and 10% for testing. This is the standard protocol in machine leaning but it assumes that the sampling has been uniformly performed on the whole corpus, which is not the case.

[4] `https://www.nb.no/forskning/nb-kilder/`
[5] `https://www.ibsen.uio.no/`
[6] `https://emunch.no/english.xhtml`
[7] An example of this is Anker & Schjønsby's *Lyset i flatene: i arkivet etter Harriet Backer* (2021).

Table 1. Number of pages by writers in train, validation and test sets for the Random split and the Writer split.

Writer	Lifespan	Random split			Writer split		
		train	val	test	train	val	test
Backer, Harriet	1845-1932	58	9	10	58	9	0
Bonnevie, Kristine	1872-1948	43	5	5	43	5	0
Broch, Lagertha	1864-1952	43			43		
Collett, Camilla	1813-1895	68	10	10	68	10	0
Garborg, Hulda	1862-1934	166	30	16	166	30	0
Hertzberg, Ebbe	1847-1912	48	6	6	48	6	0
Ibsen, Henrik	1828-1906	42	4	5	42	4	0
Kielland, Kitty	1843-1914	34	5	5	0	0	44
Munch, Edvard	1863-1944	33	5	5	0	0	43
Nielsen, Petronelle	1797-1886	58			58		
Thiis, Jens	1870-1942	41	4	4	41	4	0
Undset, Sigrid	1882-1949	40	5	5	0	0	50
Total		674	83	71	567	68	137

Table 2. Count of pages, lines, words and characters in the dataset. (Vertical text lines were ignored).

	Pages	Lines	Words	Chars
Train set	674	19,653	139,205	637,689
Validation set	83	2,286	13,916	61,560
Test set	71	1,793	11,801	52,831
Total	828	23,732	164,922	752,080

Writer split : we chose three writers that had the lowest number of pages in the train set and moved all of their pages to the test set (Kielland, Munch and Undset). In addition, we removed all the other writers from the test set. In the end, as it can be seen in table 1 there are 16% fewer pages in the train set compared to the random split. This split allows estimating the generalization capacity of the models to unseen writers.

3.3 Transcription Process

The transcriptions in the dataset were initially produced by matching existing transcriptions to the images (text-to-image), or by using pre-existing or self-trained HTR models in Transkribus. This process was followed by one round of proofreading. The proofreading was mostly done by students, with little or no prior experience with transcription. In the next phase of the project the transcriptions will be controlled by the project leaders in order to avoid inconsistencies and to further improve the "ground truth".

As mentioned above, the output of earlier efforts and projects has been an important source of transcriptions. Such output has made it possible to use a

functionality called text-to-image. This functionality makes it possible to align text lines and image lines.

3.4 Language

Today Norway has two written languages, Bokmål and Nynorsk. Danish was the de facto official language in Norway until at least 1814. Due to standardization efforts in the early 1700s, Dano-Norwegian (Danish used by Norwegians) and Danish remained almost identical throughout the the 19th century [19]. During the 19th century, several orthographic shifts took place e.g. shift from aa to å, and different ways of writing the letter ø (ö, ó, ø). Dano-Norwegian subsequently developed into Bokmål. Nynorsk, on the other hand, was conceived by the linguist Ivar Aasen in the mid-1850s with strong emphasis on the spoken dialects of Norway as well with the Old Norse language in mind. The existence of two parallel languages has resulted in a lively debate both before and after Norway gained full independence in 1905.

4 HTR libraries and models

4.1 Selection of the libraries

We conducted a survey of the HTR libraries used in recent scientific peer-reviewed articles published by the document processing community. We screened the papers published in conferences where most of the work on HTR are published: the International Conference on Document Analysis and Recognition (ICDAR), the International Conference on Frontiers of Handwriting Recognition (ICFHR), the International Workshop on Document Analysis Systems (DAS) and the International Conference on Pattern Recognition (ICPR) between 2019 and 2021. Our inclusion criteria in our study were as follows:

- the code of the system must be open source;
- the system must be compared to state-of-the-art systems on publicly available databases of handwritten documents in European languages.

Based on the open-source criteria, we identified the ten libraries that are presented in Table 3. We also included the HTR+, the system available in Transkribus even though it is not open-source because it is currently a standard tool in the community. We collected the following information on the libraries based on their source code repository:

- the type of deep learning framework used by the library;
- the number of commits and the number of different contributors to the source code;
- the date of the last commit.

We have selected the libraries to be evaluated according to the following criteria:

Table 3. Survey of open-source HTR libraries used in publications in major document processing conferences between 2019 and 2022.

Name	Framework	Last commit	Commits	Contrib.	Last version
Kaldi [1]	Kaldi	18/12/2021	9223	100	-
Kraken [13]	PyTorch	19/12/2021	1486	18	11/2021
PyLaia [24]	PyTorch	08/02/2021	860	4	12/2020
HTR-Flor++ [20]	TensorFlow 2	8/12/2021	280	4	10/2020
PyTorchOCR [4]	PyTorch	10/09/2021	24	1	-
VerticalAttentionOCR [5]	PyTorch	3/12/2021	21	1	-
Convolve, Attend & Spell [12]	PyTorch	24/06/2019	20	2	-
HRS[3]	TensorFlow	19/03/2021	20	2	-
ContentDistillation [11]	PyTorch	13/06/2020	3	1	-
Origaminet [28]	PyTorch	13/06/2020	2	2	-
HTR+ [17]	-	-	NA	NA	-

- number of commits: a large number of contributions to the library indicates regular updates, added features and active development;
- number of contributors: the code associated with publications is often the work of a single person, the main author of the publication. A low number of contributors can indicate a difficulty in handling the code and puts the maintenance and future development of the library at risk;
- date of the last commit: the quality and security of a software requires a follow-up of the dependencies updates and the application of security patches. The last commit must be recent.
- date of the last version: good software development practices recommend to index the stable and validated states of a software by numbered releases. The presence of a release is an indication of software quality.

Based on these criteria, we have selected the Kaldi, Kraken, PyLaia and HTR-Flor++ libraries. The other libraries have been discarded mainly for lack of contributors or updates. As previously mentioned, HTR+ has also been added to the selection because it is a *de facto* reference due to the success of the Transkribus platform.

4.2 Description of the selected libraries

Kaldi [1] is a library developed for speech recognition and adapted to HTR.

Kraken [13] is a turn-key OCR system optimized for historical and non-Latin script material. Since it was developed for the recognition of connected scripts such as Arabic, it is also suited to the recognition of handwritten cursive text.

PyLaia [24] is a deep learning toolkit for handwritten document analysis based on PyTorch. It is one of the HTR engines available in Transkribus.

HTR-Flor++ [20] is a framework for HTR that implements different state-of-the-art architectures, based on TensorFlow.

HTR+ [17] is the HTR system developed in the framework of the READ project and available in Transkribus.

4.3 Training of the models

In order to make a fair comparison, we have trained handwriting recognition models with each of the libraries following the provided documentation. A first model was trained with the default parameters, without optimization, which corresponds to a training performed by a non-expert (basic model). We then contacted the creators of libraries, when possible, and asked them for advice to improve the model obtained. We trained several models following their advice and selected the best one (expert model). We also trained PyLaia and HTR+ models using the Transkribus platform, which provides features for PyLaia that are not available in the open-source version. As HTR+ is not open-source, it is not possible to train or use it outside the Transkribus platform.

The details of the different models are as follows:

Kaldi basic : we trained a model according to the Bentham recipe provided in Kaldi source code. The text is modelled at BPE level with a ngram (n=3). The separate language model is trained only on the line transcriptions that come from the train set. As shown in [1] the Kaldi model training has two steps. At first a model is trained from the transcription and line image pairs ("flat start"). Then it is used to align the transcriptions on the line images. Finally another model is trained on these alignments. When the training is finished, only the last model is needed for inference - the "flat start" model can be discarded. The neural network is composed of 10 layers of SDNN. It has 6 convolution layers and 3 TDNN layers, with batch normalizations and ReLUs in between and an output layer with softmax. The lines input images are resized to a fixed height of 40 pixels while keeping the aspect ratio.

Kaldi expert : we increased the number of SDNN layers to 15. (4 extra convolution layers and one extra TDNN layer were added).

Kraken basic : the model was trained with `ketos` default parameters (input height 48). The model has 3 convolution and 3 LSTM layers and it uses group normalization.

Kraken expert : we were provided with a better model (by the authors of the library) that had 120 as input height. In addition, there was an issue with image preprocessing that hindered the performance. By using the binary format, this unnecessary preprocessing was turned off and the results got better. This model has 4 convolution and 3 LSTM layers. Group normalization layers have been replaced with dropout and max pooling.

PyLaia basic : the model was trained with default parameters, except for the input height that was fixed to 128, because without fixing the line height the model was not capable to learn much. The model has 4 convolution and 3 LSTM layers.

PyLaia expert : we tried to emulate the model that was used in Transkribus as closely as possible, but not all the parameters used were available in the

open source version. It has the same number of layers, but the number of features in convolution layers is different, max pooling is different and it uses batch normalization. Also, this one uses a different learning rate and has longer patience for early stopping.

HTR-Flor++ basic : we used the default model (Flor). It has 6 convolution and 2 GRU layers.

HTR-Flor++ expert : we tried other already implemented models (Bluche: HTR-Flor++ expert-a, Puigcerver: HTR-Flor++ expert-b, Puigcerver Octave CNN: HTR-Flor++ expert-c)[8]

HTR+ basic : we trained a HTR+ model in Transkribus from scratch. The parameters were not disclosed in the Transkribus interface.

HTR+ expert : we trained a model using a pre-trained model in Danish. The parameters were not disclosed in the Transkribus interface.

Training data The automatically created line polygons were very noisy, sometimes cutting too much of the text, making them almost unreadable. To deal with that, we decided to use bounding box extraction, which gave better results than polygon extraction in preliminary tests. Now the lines images can be noisy as well, sometimes containing (part of) the line above and below, but at least the text is visible.

Also, we ignore vertical lines in this evaluation. The performance on them could be measured in a future work.

The libraries need a transcriptions file that contains a link to the line image and the transcriptions or something similar. PyLaia requires the user to create a file with all the available symbols and transform the data before training. Other libraries do it automatically from the training data, which avoids issues from possibly malformed symbol files.

Connectionist temporal classification (CTC) loss [6] is the objective function that is used by the models to learn how to recognize the handwriting. Except for Kaldi model, that uses a lattice free maximum mutual information [23] as the objective function.

The models use early stopping, meaning that the training will be stopped if the model stops improving. Kaldi, however, uses a preset number of epochs and will complete them and then combine the model checkpoints of different epochs to produce a final model.

To compare the results of different models we used two metrics - character error rate (CER) and word error rate (WER). CER is the edit distance on character level between the predicted transcription and ground truth divided by the length of the ground truth transcription. WER is calculated in a similar way, but on word level.

Table 4. Comparison of the performance of the different models configurations (basic and expert) measured with Character Error Rates (CER) and Word Error Rates (WER) on the train, validation and test sets with random data split.

Model	Height	Augm.	Train		Val		Test	
			CER	WER	CER	WER	CER	WER
Kaldi basic	40	no	5.30	12.05	11.61	26.19	10.76	24.85
Kaldi expert	40	no	4.71	11.10	10.29	24.17	9.18	**22.19**
Kraken basic	48	no	51.95	76.52	64.60	89.72	64.44	89.49
Kraken expert	120	yes	0.40	1.31	12.05	30.29	12.20	31.28
PyLaia basic	128	no	1.37	4.45	11.02	28.09	10.87	27.62
PyLaia basic	128	yes	3.08	9.39	10.44	26.50	10.10	26.30
PyLaia expert	64	yes	3.73	10.66	11.70	28.90	12.75	31.12
PyLaia expert	128	yes	1.68	5.30	9.15	24.28	**8.86**	23.79
HTR-Flor++ basic	128	yes	-	-	-	-	11.49	31.59
HTR-Flor++ expert-a	128	yes	-	-	-	-	56.10	82.21
HTR-Flor++ expert-b	128	yes	-	-	-	-	12.62	32.33
HTR-Flor++ expert-c	128	yes	-	-	-	-	11.04	29.70
HTR+ basic	N/A	N/A	2.98	-	7.17	-	9.14	21.81
HTR+ expert	N/A	N/A	2.58	-	6.34	-	8.31	20.30

Table 5. Detailed analysis of the CER for different classes of characters on the different sets for the best HTR model (PyLaia expert) on the Random split and the Writer split.

		Random split					
		Lowercase	Uppercase	Digits	Special	Accents	Punctuation
Train	number	569,687	28,908	3,205	12,269	125	23,506
	CER	1.4	1.6	2.7	2.1	22.4	8.4
Validation	number	55,457	2,314	324	1,236	15	2,215
	CER	7.7	18.6	32.4	14.0	73.3	30.0
Test	number	47,516	2,085	319	1,040	20	1,851
	CER	7.7	13.4	24.8	14.5	75.0	28.2

5 Results

5.1 Random split

The first set of experiments was conducted on the Random Split. Table 4 reports the CER and WER for all the models on the train, validation and test sets.

As a first general remark, we found that the discussion with the creators of the libraries was often very beneficial to know how to properly configure the models or test parameters that could improve the results. In the case of Kraken, the results were very bad with the default architecture and the advice of the library experts allowed us to obtain competitive results. In the case of the other

[8] The model architectures can be seen here: `https://github.com/arthurflor23/handwritten-text-recognition/tree/master/doc/arch`

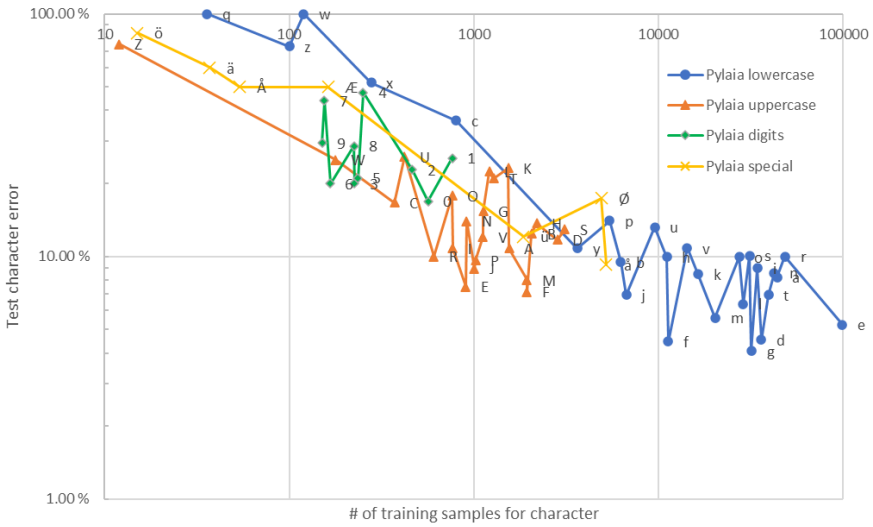

Fig. 2. Character Error Rate (CER) on the test set with respect to the number of training samples for each character, for the best HTR model (PyLaia) on the Random Split.

libraries, the discussion with the experts allowed us to validate that we had the optimal configuration.

The best CER on the test set was obtained with PyLaia using the optimized architecture with 128 pixel line height and data augmentation. The second best CER on test set was obtained by Kaldi using the optimized architecture, which also yielded the best WER. The better results of Kaldi in terms of WER can be explained by the fact that it uses an explicit language model (ngram of BPE), while the other systems model the dependencies between characters with recurrent neural networks.

It can also be noted that other systems perform better on line images with higher resolution (line height). This is especially true for Kraken which performs very poorly with the default height of 48 pixels.

The impact of data augmentation can be observed on the results of the PyLaia basic model. With data augmentation, over-training is reduced, the error rate in learning increases from 1.37% to 3.08% but the error rates in validation and testing decrease.

It should be noted that the Kaldi systems seem to suffer less from over-training than the other systems: their error rate on the training set is always higher but their error rates on the validation and test sets are among the best. This can be attributed to the use of language models by Kaldi system.

Finally, the HTR+ expert model, based on a Danish model, outperformed all the other systems. However its results are not directly comparable since this model has not been trained on exactly in the same conditions: both the line

Table 6. Detailed analysis of the confusion between characters for the best HTR model (PyLaia expert) on the Random split.

Char	# Confusions	Relative confusion	Conf. 1		Conf. 2		Conf. 3		Others
a	271	7.38 %	o	2.9 %	e	1.93 %	æ	0.79 %	1.77 %
b	42	8.08 %	l	2.9 %	t	1.54 %	h	1.35 %	2.31 %
e	207	2.60 %	a	0.5 %	o	0.39 %	i	0.29 %	1.46 %
h	86	8.13 %	s	2.5 %	t	1.13 %	k	0.85 %	3.69 %
m	74	4.49 %	n	2.61 %	v	0.61 %	i	0.24 %	1.03 %
n	189	5.59 %	r	1.72 %	m	1.18 %	v	0.68 %	2.01 %
o	162	7.98 %	a	3.20 %	e	1.87 %	ø	1.04 %	1.87 %
r	198	5.18 %	s	0.89 %	n	0.89 %	v	0.55 %	2.85 %
s	188	7.25 %	r	1.74 %	h	1.04 %	e	0.81 %	3.66 %
F	5	5.21 %	T	2.1 %	f	1.04 %	d	1.04 %	1.04 %
L	13	20.00 %	t	9.2 %	l	3.08 %	d	3.08 %	4.62 %
æ	34	7.93 %	e	2.3 %	a	2.10 %	d	0.93 %	2.56 %
ø	56	14.74 %	o	6.1 %	å	2.37 %	e	1.58 %	4.74 %
å	21	11.60 %	ø	4.4 %	a	3.32 %	u	1.11 %	2.76 %

extraction and the evaluation were done in Transkribus and are not open-source, so they may be different from our line extraction and CER/WER metrics.

We conducted a detailed analysis of the CER for each characters with respect to their number of training samples, presented on Figure 2. Lower case letters are by far the majority and therefore the best recognized, except for some rare letters (*qzwx*). Numbers and special characters are very poorly represented (except å and ø) and therefore very poorly recognized. Capital letters are in a intermediate situation, with a relatively small number of samples but relatively low error rates. A summary of the CER for different classes of characters is presented on Table 5. One can note that punctuation is particularly difficult to recognize: even with almost as many examples as capital letters, their error rate is twice as high.

Finally, we analyzed the most frequent confusions between characters on the test set for the best HTR model (PyLaia expert). An extract of the confusion table is presented in Table 6. Compared with results from printed OCR [21], the HTR confusions are more spread across confusion alternatives and the typical OCR single character substitutions like e-o-c, h-b, n-u, m-n, i-l-I while present in the HTR results, have less relative weight. The same is true for the typical OCR character substitutions for the Norwegian special characters: ø-o, æ- a or e, å - a, which are also present in the HTR with less relative weight while some confusion not often seen in OCR such as å-i, ø-, æ-o and å-r are relatively important. OCR errors are generally caused by noise and low contrast where the basic characters are in essence very similar, but HTR errors are generally caused by different graphical representations of different characters most prominent when comparing different writers, but there might also be large differences of character representations for a single writer. Some common characters, e.g. 'a','g' and 'r' have generally different topology in cursive handwriting compared to print which also affects the confusion alternatives.

Table 7. Comparison of the performance of the different configuration (basic and expert) of Kaldi and PyLaia models measured with Character Error Rates (CER) and Word Error Rates (WER) on the train, validation and test sets with the Writer split.

Model	Height	Augm.	Train		Val		Test	
			CER	WER	CER	WER	CER	WER
Kaldi basic	40	no	4.90	11.34	12.57	28.10	24.24	44.49
Kaldi expert	40	no	4.37	10.48	11.03	25.79	**21.79**	**42.13**
PyLaia basic	128	yes	2.70	8.25	10.64	27.58	24.36	49.42
PyLaia expert	128	yes	1.64	5.40	9.53	25.90	22.74	47.95

5.2 Random split by writer with unseen writers

We chose the best models from the previous experiments (PyLaia and Kaldi) and trained them on the Writer split. This experiment allows us to evaluate the generalization capabilities of the different models to new writers. As it might be expected, the models perform a lot worse on unseen writers. PyLaia and Kaldi obtain very similar results, with a slight advantage to Kaldi. Again, this advantage can be attributed to the use of a language model by Kaldi but also to the lower resolution of the input images, which may reduce over-training.

6 Conclusion

In this article we have introduced a database of handwritten historical documents in Norwegian. This database is the first of its kind and constitutes a valuable resource for the development of handwritten text recognition models (HTR) in Norwegian. In order to evaluate the performance of state-of-the-art HTR models on this new database, we conducted a systematic survey of open-source HTR libraries published between 2019 and 2021. We selected four libraries, amongst ten, according to criteria of quality and sustainability of their source code based on software development metrics. We trained twelve models in different configurations and compared their performance on both random data splitting and writer-based data splitting to evaluate their generalization capabilities to writers not seen during training. Finally, we studied the most frequent confusions between characters. The best recognition results were obtained by the Kaldi library which uses a language model and PyLaia which uses higher resolution images and data augmentation during training. A combination of these different techniques, in a single model or by voting, should further increase the performance of HTR models. Recently proposed models based on transformers[15] should also be added to the benchmark.

7 Acknowledgements

We thank Daniel Stoekl, Benjamin Kiessling, Joan Andreu Sanchez, Arthur Flôr for their advices during the training of the HTR models with their respecting

libraries. This work was supported by the Research Council of Norway through the 328598 IKTPLUSS HuginMunin project.

References

1. Arora, A., Chang, C.C., Rekabdar, B., BabaAli, B., Povey, D., Etter, D., Raj, D., Hadian, H., Trmal, J., Garcia, P., et al.: Using ASR methods for OCR. In: International Conference on Document Analysis and Recognition (2019)
2. Augustin, E., Brodin, J.M., Carré, M., Geoffrois, E., Grosicki, E., Prêteux, F.: RIMES evaluation campaign for handwritten mail processing. International Conference on Document Analysis and Recognition p. 5 (2006)
3. Chammas, E., Mokbel, C., Likforman-Sulem, L.: Handwriting recognition of historical documents with few labeled data. In: International Workshop on Document Analysis Systems. pp. 43–48. IEEE (2018)
4. Coquenet, D., Chatelain, C., Paquet, T.: Recurrence-free unconstrained handwritten text recognition using gated fully convolutional network. In: International Conference on Frontiers in Handwriting Recognition. pp. 19–24 (2020)
5. Coquenet, D., Chatelain, C., Paquet, T.: End-to-end handwritten paragraph text recognition using a vertical attention network. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
6. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: international conference on Machine learning. pp. 369–376 (2006)
7. Hegghammer, T.: OCR with tesseract, amazon textract, and google document AI: a benchmarking experiment. Journal of Computational Social Science (2021)
8. Hodel, T., Schoch, D., Schneider, C., Purcell, J.: General models for handwritten text recognition: Feasibility and state-of-the art. german kurrent as an example. Journal of Open Humanities Data **7** (2021)
9. Hussain, R., Raza, A., Siddiqi, I., Khurshid, K., Djeddi, C.: A comprehensive survey of handwritten document. Journal on Advances in Signal Processing **2015**, 46 (2015)
10. Jørgensen, F., Aasmoe, T., Ruud Husevåg, A.S., Øvrelid, L., Velldal, E.: NorNE: Annotating named entities for Norwegian. In: Language Resources and Evaluation Conference (2020)
11. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Distilling content from style for handwritten word recognition. In: International Conference on Frontiers in Handwriting Recognition (2020)
12. Kang, L., Toledo, J.I., Riba, P., Villegas, M., Fornés, A., Rusiñol, M.: Convolve, attend and spell: An attention-based sequence-to-sequence model for handwritten word recognition. In: German Conference for Pattern Recognition (2018)
13. Kiessling, B., Tissot, R., Stokes, P., Stökl Ben Ezra, D.: eScriptorium: An Open Source Platform for Historical Document Analysis. In: International Conference on Document Analysis and Recognition Workshops. vol. 2, pp. 19–19 (2019)
14. Kummervold, P.E., de la Rosa, J., Wetjen, F., Brygfjeld, S.A.: Operationalizing a national digital library: The case for a norwegian transformer model. In: Nordic Conference on Computational Linguistics (2021)
15. Li, M., Lv, T., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: Trocr: Transformer-based optical character recognition with pre-trained models (2021), https://arxiv.org/abs/2109.10282

16. 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)
17. Michael, J., Weidemann, M., Labahn, R.: Htr engine based on nns p 3 optimizing speed and performance-htr +. Tech. rep., READ - H2020 Project 674943 (2018)
18. Muehlberger, G., Seaward, L., Terras, M., Oliveira, S., Bosch, V., Bryan, M., Colutto, S., Déjean, H., Diem, M., Fiel, S., Gatos, B., Grüning, T., Greinoecker, A., Hackl, G., Haukkovaara, V., Heyer, G., Hirvonen, L., Hodel, T., Jokinen, M., Zagoris, K.: Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study. Journal of Documentation (2019)
19. Nesse, A., Sandøy, H.: Norsk språkhistorie: Tidslinjer. IV. Novus (2018)
20. Neto, A.F.S., Bezerra, B.L.D., Toselli, A.H., Lima, E.B.: HTR-Flor++: a handwritten text recognition system based on a pipeline of optical and language models. In: ACM Symposium on Document Engineering (2020)
21. Nguyen, T.T.H., Jatowt, A., Coustaty, M., Nguyen, N.V., Doucet, A.: Deep Statistical Analysis of OCR Errors for Effective Post-OCR Processing. In: Joint Conference on Digital Libraries (2019)
22. Ortiz, P., Burud, S.: Bert attends the conversation: Improving low-resource conversational asr (2021), `https://arxiv.org/abs/2110.02267`
23. Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., Khudanpur, S.: Purely sequence-trained neural networks for asr based on lattice-free mmi. In: Interspeech. pp. 2751–2755 (2016)
24. Puigcerver, J., Mocholí, C.: Pylaia. `https://github.com/jpuigcerver/PyLaia` (2018)
25. Strauß, T., Leifert, G., Labahn, R., Mühlberger, G.: Competition on automated text recognition on a read dataset. In: International Conference on Frontiers in Handwriting Recognition (2018)
26. Sánchez, J.A., Romero, V., Toselli, A., Villegas, M., Vidal, E.: A set of benchmarks for handwritten text recognition on historical documents. Pattern Recognition **94** (2019)
27. Toiganbayeva, N., Kasem, M., Abdimanap, G., Bostanbekov, K., Abdallah, A., Alimova, A., Nurseitov, D.: KOHTD: kazakh offline handwritten text dataset (2021), `https://arxiv.org/abs/2110.04075`
28. Yousef, M., Bishop, T.E.: Origaminet: Weakly-supervised, segmentation-free, one-step, full page textrecognition by learning to unfold. In: Conference on Computer Vision and Pattern Recognition (2020)