

# Improving OCR Accuracy on Early Printed Books by combining Pretraining, Voting, and Active Learning

---

## Abstract

We combine three methods which significantly improve the OCR accuracy of OCR models trained on early printed books: (1) The pretraining method utilizes the information stored in already existing models trained on a variety of typesets (*mixed models*) instead of starting the training from scratch. (2) Performing cross fold training on a single set of ground truth data (line images and their transcriptions) with a single OCR engine (OCRopus) produces a committee whose members then vote for the best outcome by also taking the top-N alternatives and their intrinsic confidence values into account. (3) Following the principle of maximal disagreement we select additional training lines which the voters disagree most on, expecting them to offer the highest information gain for a subsequent training (active learning). Evaluations on six early printed books yielded the following results: On average the combination of pretraining and voting improved the character accuracy by 46% when training five folds starting from the same mixed model. This number rose to 53% when using different models for pretraining, underlining the importance of diverse voters. Incorporating active learning improved the obtained results by another 16% on average (evaluated on three of the six books). Overall, the proposed methods lead to an average error rate of 2.5% when training on only 60 lines. Using a substantial ground truth pool of 1,000 lines brought the error rate down even further to less than 1% on average.

## 1 Introduction

Recent progress on OCR methods using recurrent neural networks with LSTM architecture (Hochreiter and Schmidhuber, 1997) enabled effective training of recognition models for both modern (20th century and later) and historical (19th century and earlier) manuscripts and printings (Fischer et al., 2009; Breuel et al., 2013; Springmann et al., 2014; Springmann and Lüdelling, 2017). Individually trained models regularly reached character recognition rates of about 98% for even the earliest printed books. The need to train individual models in order to reach this level of recognition accuracy for early printings sets the field of historical OCR apart from readily available (commercial and open-source) *general* (also called *polyfont* or *omnifont*) models trained on thousands of modern fonts which yield better than 98% recognition rates on 19th century printings and better than 99% on modern documents. Training historical recognition models on a variety of typesets results in *mixed models* which may be seen as a first approximation

to modern polyfont models, but their predictive power is considerably lower than that of individual models.

In view of the mass of available scans of historical printings we clearly need automatic methods of OCR which in turn require good historical polyfont models. As long as these models are not available and at present cannot be easily constructed (we lack the necessary historical fonts to be able to synthesize large amounts of training material automatically), our next best approach is to maximize the recognition rate of a small amount of manually prepared ground truth (GT). This is the subject of the present paper which applies the methods of pretraining, voting, and active learning (AL) to the field of historical OCR. Using an already trained model as a starting point for subsequent training with additional individual material requires the capability to add specific characters not previously included in the symbol set (the *codec*) and the dynamic expansion (and reduction) of the output layer of the neural network. In the context of recurrent neural networks this was recently made possible by Christoph Wick<sup>1</sup> as reported in Reul et al. (2017c). Voting is a well known method of classifier combination resulting in fewer errors than the best single classifier output (Rice et al., 1992). Active learning ensures that lines showing maximal disagreement among classifiers are included in the training set to enable the maximal learning effect. While more training data is always better, combining these three methods results in a level of recognition accuracy that could otherwise only be reached by a much larger amount of GT and therefore a much larger manual effort to generate it.

Section 2 summarizes the extensive corpus of related work for each of the three methods. In Section 3 we describe the printing material which the experiments of Section 4 are based on. Section 5 contains the discussion of our results and we conclude the paper with Section 6.

## 2 Related Work

In this section we first sum up a selection of important contributions concerning OCR relevant to our task and introduce findings with regard to training and applying mixed models using OCRopus. Next, a brief summary of the history of voting techniques and applications in the field of OCR is provided. After a short section on transfer learning we give an overview over some basic AL concepts.

### 2.1 OCR and Mixed Models

Breuel et al. (2013) used their own open source tool OCRopus<sup>2</sup> to recognize modern English text and German Fraktur from the 19th century by training mixed models, i.e. models trained on a variety of fonts, typesets, and interword distances from different books. The English model was trained on the UW-III data set<sup>3</sup> consisting of modern

<sup>1</sup>[https://github.com/ChWick/ocropy/tree/codec\\_resize](https://github.com/ChWick/ocropy/tree/codec_resize)

<sup>2</sup><https://github.com/tmbdev/ocropy>

<sup>3</sup><http://isis-data.science.uva.nl/events/dlia/datasets/uwash3.html>

English prints. Applying the model to previously unseen lines from the same dataset yielded a character error rate (CER) of 0.6%. The training set for the Fraktur model consisted of mostly synthetically generated and artificially degraded text lines. The resulting model was evaluated on two books of different scan qualities yielding CERs of 0.15% and 1.37%, respectively.

Ul-Hasan and Breuel (2013) promoted an approach not only mixing different types but also various languages by generating synthetic data for English, German, and French. Apart from three language specific models they also trained a mixed one. While the language specific models unsurprisingly performed best when applied to test data of the same language yielding CERs of 0.5% (English), 0.85% (German) and 1.1% (French) the mixed model also achieved a very low CER of 1.1% on a mixed dataset. These experiments indicate a certain robustness or even independence of the OCRopus LSTM architecture regarding different languages in mixed models.

After proving that OCR on even the earliest printed books is not only possible but can be very precise (down to 1% error rate, Springmann, 2015), Springmann et al. adapted the idea of training mixed models to early prints in different application scenarios. In Springmann et al. (2016) their corpus consisted of twelve Latin books printed with Antiqua types between 1471 and 1686. Training on one half of the books and evaluating on the other half mostly yielded CERs of under 10%. Admittedly, these results are far off the numbers reported above which can be explained by the vastly increased variety of the types. Still, the trained models provide a valid starting point for further model improvements through individual training. Additionally, a clear correlation between the intrinsic confidence values of OCRopus and the resulting CER was demonstrated.

In Springmann and Lüdeling (2017) a similar experiment was conducted on the 20 German books of the RIDGES Fraktur corpus<sup>4</sup>. Again, by training mixed models on half of the books and evaluating on the held-out data impressive recognition results of around 5% CER on average were achieved. As expected, the individually trained models performed even better, reaching an average CER of around 2%.

## 2.2 Alignment and Voting

Handley (1998) gives an overview regarding topics concerning the improvement of OCR accuracy through the combination of classifier results and discusses different methods to combine classifiers and string alignment approaches.

Rice and Nartker (1996) released a collection of command line scripts for the evaluation of OCR results called the ISRI Analytic Tools. Their voting procedure first aligns several outputs using the Longest Common Substring (LCS) algorithm (Rice et al., 1994) and then performs a majority vote. In several competitions they applied their tools to evaluate the results of various commercial OCR engines on modern prints (see, e.g., Rice et al., 1992, 1996). By voting on the output of five engines on English business letters the character accuracy rate ( $CAR = 1 - CER$ ) increased from between 90.10% and 98.83% to 99.15%.

---

<sup>4</sup><http://korpling.org/ridges>

A simple but effective way to achieve variance between the voting inputs was proposed by Lopresti and Zhou (1997) by simply scanning each page three times. While using only a single OCR engine they still achieved a reduction of error rates between 20% and 50% on modern prints resulting in a CAR of up to 99.8%.

Boschetti et al. (2009) improved the output of the best single engine (ABBYY, up to 97% CAR) by an absolute value of 2.59 percentage points by applying a Naive Bayes classifier on the aligned output of three different engines on Ancient Greek editions from the 19th and 20th century. Beforehand, they performed a progressive alignment which starts with the two most similar sequences and extends the alignment by adding additional sequences.

Lund et al. (2011) used voting and dictionary features as well as maximum entropy models trained on synthetic data. Applied to a collection of typewritten documents from Word War II they recorded a relative gain of 24.6% over the word error rate of the best of the five employed OCR engines.

An approach for aligning and combining different OCR outputs applicable to entire books was introduced by Wemhoener et al. (2013). First, a pivot is chosen among the outputs. Then, all other outputs are aligned pairwise with the pivot by first finding unique matching words in the text pairs to align them using an LCS algorithm. By repeating this procedure recursively, two texts can be matched in an efficient way. Finally, all pairs are aligned along the pivot and a majority vote determines the final result.

Liwicki et al. (2011) tackled the task of handwritten text recognition acquired from a whiteboard by combining several individual classifiers of diverse nature. They used two base recognizers which incorporated hidden Markov models and bidirectional LSTM networks and trained them on different feature sets. Moreover, two commercial recognition systems were added to the voting. The multiple classifier system reached an accuracy of 86.16% on word level and therefore outperformed even the best individual system (81.26%) significantly.

Al Azawi et al. (2015) trained neural LSTM networks on two OCR outputs aligned by weighted finite-state transducers based on edit rules in order to return a best voting. After training the network on a vast amount of data very similar to the test set, it was able to predict even characters which were not correctly recognized by either of the two engines. During tests on printings with German Fraktur and the UW-III data set the LSTM approach led to CERs of around 0.40%, considerably outperforming the ISRI voting tool and the method presented in Wemhoener et al. (2013) (between 1.26% and 2.31%). However, applying this method to historical spellings has a principal drawback as it relies on fixed input-output relationships. Since historical spelling patterns are much more variable than modern ones and the same word is often spelled and printed in more than one form even in the same document, it is not possible or at least may not be desired to map each OCR token to a single ‘correct’ token.

In Reul et al. (2018) we implemented a cross-fold training procedure with subsequent confidence voting in order to reduce the CER on early printed books. This method shows considerable differences compared to the work presented above. Not only is it

applicable to some of the earliest printed books, but it also works with only a single open source OCR engine. Furthermore, it can be easily adapted to practically any given book using even a small amount of GT without the need for excessive data to train on (60 to 150 lines of GT corresponding to just a few pages will suffice for most cases).

By dividing the GT into  $N$  different folds and aligning them in a certain way we were able to train  $N$  strong but also diverse models. Then, these models acted as voters both in the default sequence voting (see ISRI tools above) and a newly created confidence voting scheme which also takes the intrinsic confidence information of the top- $n$  (not just top-1) predictions of OCRopus into consideration. Experiments on seven books printed between 1476 and 1675 led to the following observations:

1. For all experiments the cross fold training and voting approach led to significantly lower CERs compared to performing only a single training. Gains between 19% and 53% were reported for several books and different number of lines of GT.
2. OCR texts with a lower CER benefitted even more than more erroneous results.
3. The amount of available GT did not show a notable influence on the improvements achievable by confidence voting. Yet, a very high number of lines leads to a drop in voting gains for most books. This has to be expected for models that get closer to perfection as most of the remaining errors are unavoidable ones such as characters with defects or untrained glyphs missing in the training set.
4. Increasing the number of folds can bring down the CER even further, especially when training on a large set of lines. However, considering the range of available GT lines and the required computational effort, five folds appeared to be a sensible default choice until further testing regarding parameter optimization has been performed.
5. The confidence voting always outperformed the standard sequence voting approach by reducing the amount of errors by another 5% to 10%.

### 2.3 Transfer Learning and OCR Pretraining

While to the best of our knowledge there is no suitable related work regarding transfer learning in the field of OCR, it was applied successfully to a variety of other tasks (e.g. Yosinski et al. (2014) for labeling arbitrary images and Wick and Puppe (2017) for leaf classification) by deploying deep convolutional neural networks after performing a pretraining on data from a different but somewhat similar recognition task.

Admittedly, these examples of transfer learning used far deeper networks than OCRopus with only a single hidden layer, resulting in a dramatically increased number of parameters and consequently more opportunities to learn and remember useful low-level features. Nonetheless, since scripts in general should show a high degree of similarity we still expected a noteworthy impact of pretraining and studied the effect of building from an already available mixed model instead of starting training from scratch

(see Reul et al., 2017c). As starting points we used the models for modern English, German Fraktur from the 19th century, and the Latin Antiqua model described above. From our experiments we arrived at the following conclusions:

1. Building from a pretrained model significantly reduced the obtainable CER compared to starting the training from scratch.
2. Improvement rates decrease with an increasing amount of GT lines available for training. While models trained on only 60 lines of GT gained over 40% on average over starting from scratch, this number went down to around 15% for 250 lines.
3. The incorporation of a whitelist for standard letters and digits which cannot be deleted from the codec even if they do not occur in the training GT showed an additional average gain of 5%.
4. Even the mixed models for modern English and 19th century Fraktur which were completely unrelated to the individual books in terms of printing type and age of the training material led to significant improvements compared to training from scratch.

## 2.4 Active Learning

Settles (2012) gives a very comprehensive overview over the literature dealing with Active Learning (AL). Apart from introducing different usage scenarios and discussing theoretical and empirical evidence for the application of AL techniques they define a typical AL scenario as follows: A learner starts out with access to a (possibly very small) pool of labeled examples to learn from. In order to improve performance it is possible to send queries consisting of one or several carefully selected unlabeled examples to a so-called oracle (teacher/human annotator) who then returns a label for each example in the query. Afterwards, the learner can utilize the obtained additional data. Obviously, the progress of the learner heavily depends on the examples selected to be labeled. Furthermore, the goal is to learn as much as possible from as few as possible queried examples, keeping the oracle's effort to a minimum.

One of the most successful query techniques is called *query by committee* and was introduced by Seung et al. (1992). The basic idea is that a committee of learners/models/voters is trained on the current labeled set. Each member of the committee is allowed to cast a vote on a set of query candidates (unlabeled examples). The assumption is that the candidate the voters disagree most on is also the one which offers the biggest information gain when being added to the training set. This is called the *principle of maximal disagreement*.

Among others, the effect of this approach was demonstrated by Krogh and Vedelsby (1995) who trained five neural networks to approximate the square-wave-function. They performed 2x40 independent test runs starting from a single example and using passive and active learning. While the next example was chosen randomly during the passive tests the networks always got handed the example with the largest ambiguity among

**Table 1:** Books used during the experiments as well as the amount of GT lines set aside for Training, Evaluation, and Active Learning.

ID/Year	Language	Training	Evaluation	Active Learning
1476	German	1,000	1,000	750
1488	German	1,000	1,000	1,928
1505	Latin	1,000	1,000	1,039
1495	German	1,000	1,114	-
1500	Dutch	1,000	1,250	-
1572	Latin	1,000	1,098	-

the five voters out of 800 random ones. Evaluation showed that AL led to a significantly better generalization error and that the individual additional training examples on average contributed much more to the training process when chosen according to the principle of maximal disagreement.

As for OCR, Springmann et al. (2016) performed some initial experiments on selecting additional training lines in an active way. After recognizing lines with a mixed model they tested several strategies according to which they chose lines for further transcription. The best result was obtained when using a mixture of randomly selected lines combined with lines with low confidence values. It is worth mentioning that after transcribing these lines they started their training from scratch since the pretraining approach introduced above had not been developed, yet.

### 3 Materials and Methods

In this section we first introduce the early printed books and mixed models we used for our experiments. Then our previous approaches for separate voting and pretraining are briefly described on a technical level. Finally, we show how the principle of maximal disagreement can be utilized in order to choose additional training lines in an informed way within an iterative AL approach.

#### 3.1 Books

The experiments were performed on six early printed books (see Table 1). The AL experiments were carried out on the three books above the horizontal line. We focused on these books as they provided a large amount of GT which is needed to perform the procedure. In a real world application scenario it would be sensible to choose the additional training lines by recognizing all lines without GT and choose the worst ones. Therefore, as many lines as possible are required to be able to evaluate this scenario.

To avoid unwanted side effects resulting from different types or varying line length only lines from running text were used and headings, marginalia, page numbers, etc. were excluded. 1505 represents an exception to that rule as we chose the extensive

Dem hoeghen marschalck sent qurijn  
 dienen. Da was der keyser gar fro dz sein syn  
 en vnd materien erlengert vnd sch  
 Sonder beghinzel Dorfmarct in Vreughden.  
 Stultoz ꝛc. Qm̄ inquā stultoz vt  
 complexus fit: ut sub æquatore est

**Figure 1:** Different example lines from the six books used for evaluation. From top to bottom: excerpts from books 1476, 1488, 1495, 1500, 1505, and 1572.

commentary lines instead, as they presented a bigger challenge due to very small inter character distances and a higher degree of degradation. Figure 1 shows some example lines.

The books published in 1495, 1500, and 1505 are editions of the Ship of Fools *Narrenschiff* and were digitized as part of an effort to support the Narragonien project at the University of Würzburg<sup>5</sup>. Despite their similar content these books differ considerably from an OCR point of view since they were printed in different print shops using different typefaces and languages (Latin, German, and Dutch, see Figure 1). Ground truth for the 1488 book was gathered during a case study of highly automated layout analysis (see Reul et al., 2017a). 1476 is part of the Early New High German Reference Corpus<sup>6</sup> and 1572 was digitized in order to be added to the AL-Corpus<sup>7</sup> of Arabic-Latin translations.

### 3.2 Mixed Models

During our experiments we made use of three mixed models. Our first model LH (abbreviated for Latin Historical) was trained on all twelve historical books introduced in Springmann et al. (2016). The books are printed in Latin and with Antiqua types. The training was performed on 8,684 lines and was stopped after 109,000 iterations. We evaluated all resulting models on 2,432 previously unseen test lines in order to determine the best model which occurred after 98,000 training steps achieving a CER of 2.92% .

<sup>5</sup><http://kallimachos.de/kallimachos/index.php/Narragonien>

<sup>6</sup><http://www.ruhr-uni-bochum.de/wegera/ref/index.htm>

<sup>7</sup><http://arabic-latin-corpus.philosophie.uni-wuerzburg.de>

Additionally, we used the freely available OCRopus standard models for English (ENG)<sup>8</sup> and German Fraktur (FRK)<sup>9</sup> introduced in Breuel et al. (2013) and described above.

It is worth mentioning that all the books, which were introduced in the last section and will be used for evaluation, were disjoint with the training materials of the mixed models.

### 3.3 Cross Fold Training and Confidence Voting

In the absence of viable alternatives to OCRopus we introduced variations in the training data in order to obtain highly performant individual yet diverse voters. This was done by applying a technique called cross fold training: The available GT is divided into  $N$  folds with  $N$  being the number of models which will later participate in the vote. Then,  $N$  training processes take place by using one fold for testing, i.e. choosing the best model, and the rest for training. While the training data shows a significant overlap for each training the folds used for testing are distinct.

After determining the best model of each training process each one of the resulting best models recognizes the same set of previously unknown lines. As an output not only the OCR text is stored but also so-called *extended lloc* (**L**STM **l**ocation **o**f **c**haracters) files which store the probability of the actually recognized character as well as the probabilities of its alternatives.

During the confidence voting process the different result texts are first aligned by applying the ISRI sync tool which identifies the positions of OCR differences as well as the output of each voter at this position. The confidence voting is performed by identifying the corresponding extended llocs and adding up the confidence values for each of the character alternatives. The character with the highest confidence sum gets voted into the final output (Reul et al., 2018).

### 3.4 Pretraining utilizing Transfer Learning

The character set of pretrained mixed models often comprises more or fewer characters than the data the new model is supposed to be trained on. In order to assure full compatibility we had to make some enhancements on the code level regarding the OCRopus training process which are available at GitHub<sup>10</sup>. These changes allow us to comfortably extend or reduce the codec depending on the available training GT and the characters it contains. When starting the training process the character sets of the existing model and the GT are matched. For each character which occurs in the GT but is not part of the model an additional row is attached to the weight matrix and the corresponding weights are randomly initialized. A character which is known to the model but is not part of the GT leads to the deletion of the corresponding row in the

---

<sup>8</sup><http://www.tmbdev.net/en-default.pyrnn.gz>

<sup>9</sup><http://tmbdev.net/ocropy/fraktur.pyrnn.gz>

<sup>10</sup>[https://github.com/ChWick/ocropy/tree/codec\\_resize](https://github.com/ChWick/ocropy/tree/codec_resize)

matrix. In order to avoid blind spots especially when dealing with small amounts of GT and important but less frequent characters like numbers or capital letters it is possible to define a so-called *whitelist*. Characters on the whitelist will not get deleted from the matrix regardless of whether they occur within the GT or not (Reul et al., 2017c).

### 3.5 Active Learning

As explained above the basic idea behind Active Learning is to allow the learners, i.e. the different voters (see 3.5), to decide which training examples they benefit the most from instead of selecting additional lines randomly. Since in a real world application scenario there usually is no GT available for potential new training lines, we cannot just use the ones for which our current models give the worst results, i.e. the ones with the highest CER. However, we can still identify the lines we expect to be most suitable for further training by following the principle of maximal disagreement.

After recognizing a line with each model of an ensemble consisting of  $n$  voters, all outputs are compared to each other in pairs. As a measure for the difference between two OCR output strings  $a$ ,  $b$ , we define the Levenshtein Distance Ratio  $LDR(a, b)$  as the Levenshtein distance between  $a$  and  $b$ , divided by the maximum string length of  $a$  and  $b$ . These ratios are then summed up between all pairs of different OCR output strings and divided by the number of pairs  $n(n-1)/2$  to yield the average  $LDR_{\emptyset}$ . The pseudo-code in Algorithm 1 details our calculations.

**Data:** *line* image without GT, ensemble of  $n$  voters ( $n > 1$ )

**Result:** the  $LDR_{\emptyset}$  of the *line*

*outputs*  $\leftarrow$  *recognizeWithAllVoters(line)*

*sum* = 0

**foreach**  $a_i, a_j$  *with*  $(i < j) \in$  *outputs* **do**

$LDR(a, b) \leftarrow \frac{Lev(a, b)}{\max\{|a|, |b|\}}$   
    *sum*  $\leftarrow$  *sum* +  $LDR(a, b)$

**end**

$LDR_{\emptyset} = \frac{sum}{n(n-1)/2}$

**Algorithm 1:** The calculation of a line’s average Levenshtein Distance Ratio ( $LDR_{\emptyset}$ ).  $Lev(a, b)$  denotes the Levenshtein distance between two outputs  $a$  and  $b$ . The length (= the number of characters) of an output  $a$  is given by  $|a|$ .

Finally, after processing various lines, they are sorted in descending order according to their  $LDR_{\emptyset}$ s. In a real world application scenario the lines are handed to a human annotator who then produces GT by transcribing them or by correcting one of the individual OCR results or the output of the confidence voting, if available. While the idea of the whole process is to give the committee the lines it requested, it is still up to the human annotator to decide whether a line is suitable for further training or not.

For example, a line might have been badly recognized due to a severe segmentation error or due to an unusually high degree of degradation making recognition pretty much impossible. These lines cannot be expected to make a noteworthy contribution to the training process and are therefore discarded.

### 4 Experiments

In order to evaluate the effectiveness of the methods described above we performed two main experiments. First, the voting and pretraining approaches were combined<sup>11</sup> by performing the voting procedure with models which were not trained from scratch but started from one or several pretrained mixed models. Second, the voters resulting from the first experiment served as a committee during an AL approach following the principle of maximal disagreement.

Since the train/test/evaluation distribution of the GT lines has changed, the results can differ from the ones obtained from earlier experiments. Based on the previous results reported in sections 2.2 and 2.3 we chose to implement the following guidelines for all of the upcoming experiments.

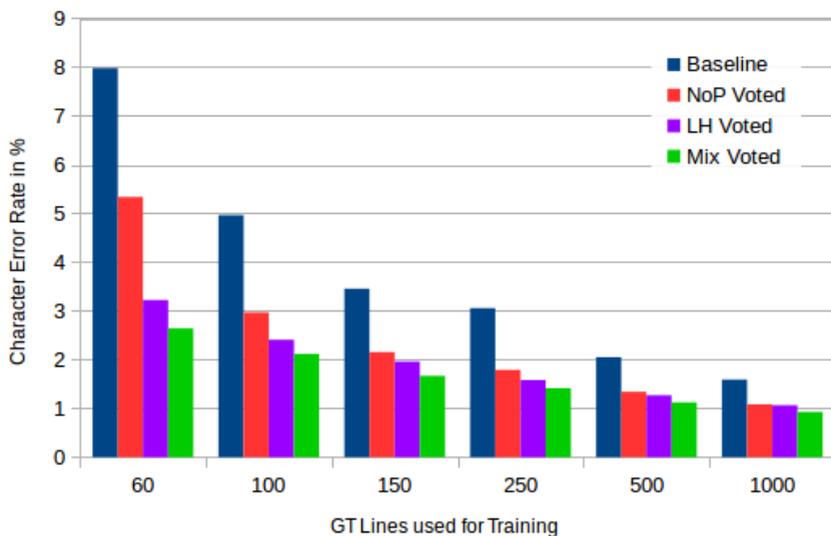
1. The number of folds during cross-fold training and consequently the number of voters is set to 5.
2. OCR results are always combined by enforcing the confidence voting approach.
3. Whenever pretraining is used a generic minimal whitelist consisting of the letters a-z, A-Z and the digits 0-9 is added to the codec.
4. Each model training is carried out until no further improvement is expected (e.g. 30,000 iterations for 1,000 lines of training GT).

#### 4.1 Combining Pretraining and Voting

Naturally, the combination of voting and pretraining seems attractive and should be evaluated. The number of lines used for training was varied in six steps from 60 to 1,000. Each set of lines was divided into five folds and the allocation was kept fixed for all experiments. We used two different approaches for pretraining. First, we trained the five voters by always building from the LH model since it yielded the best results during previous experiments. Second, we varied the models used for pretraining. We kept voters 1 and 2 from the first setup (trained from LH). For voters 3 and 4 FRK was utilized as a starting point since it was trained on German Fraktur fonts which are somewhat similar to the broken script of the books at hand. Only one voter (5) was built from ENG as it was the least similar one out of the available mixed models regarding both age and printing type of the training data. This setup was slightly

---

<sup>11</sup>The corresponding code is available at GitHub: <https://github.com/chreul/mptv>



**Figure 2:** Comparison of the CERs (averaged over all books for each set of lines) of four different approaches: *Baseline* (no pretraining, no voting), *NoP* (no pretraining, voted), *LH* (all five folds trained from the LH model, voted), and *Mix* (mixed pretraining, voted).

adapted for book 1572 as it was printed using Antiqua types. Therefore, in this case one of the two FRK folds was pretrained with ENG instead.

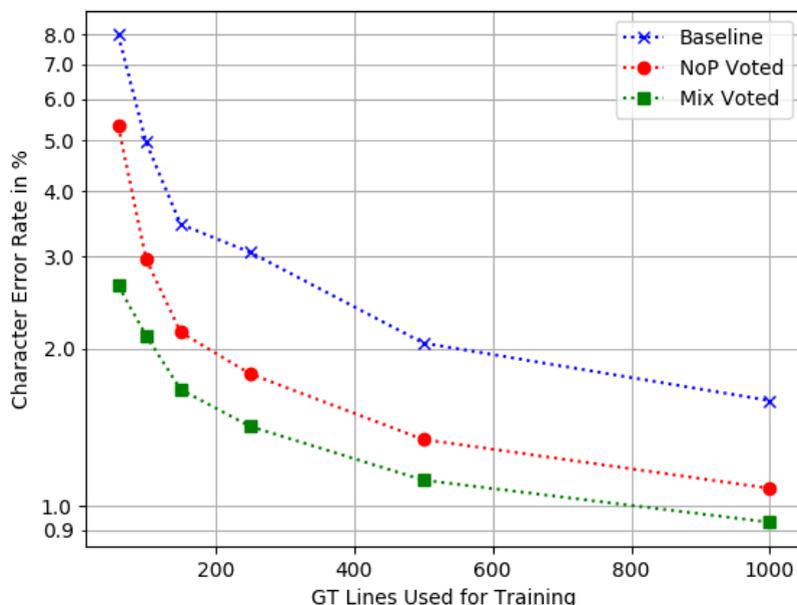
The idea was to still train strong individual models while increasing diversity among them, hoping for a positive effect on the final voting output. The results are summed up in Table 2. For reasons of clarity, detailed numbers are only provided for three books, i.e. the ones which will be used for further experiments. The general behaviour averaged over all books can be seen in Figure 2 and an overview over the progress made by adding more training lines is presented in Figure 3.

In the majority of cases the combination of pretraining and voting considerably outperforms both the default voting approach showing gains of 14% (LH) and 26% (Mix) as well as the default pretraining approach showing gains of 29% (LH) and 39% (Mix) when averaging over all six books and lines. As expected, the best improvements can be achieved when using a small number of GT lines, resulting in gains ranging from 40% (LH) and 51% (Mix) for 60 lines to 2% (LH) and 14% (Mix) for 1,000 lines.

Overall, the average CER on individual folds pretrained with LH (2.70%) is effectively identical to the one achieved by building from a variety of mixed models (2.69%). However, in the case of applying the voting procedure over all five folds, the Mix approach yields considerably better results than just using LH as a pretrained model,

**Table 2:** CER in % of combining pretraining (*Single Folds*) and voting (*Voting Result*). *Single Folds* contain the baseline without pretraining (*Base*), pretraining with LH model (*LH*), and with a mixture of models (LH, LH, FRK, FRK/ENG, ENG) (*Mix*). *Voting Result* shows the results of different voters based on no pretraining (*NoP*), pretraining with LH model (*LH*), and with mixed model (*Mix*). The *Improvement* columns show the voting gains of LH over NoP (*NL*), Mix over NoP (*NM*), Mix over LH (*LM*), and Mix over the base (*BM*). The underlined CERs represent the starting points for the upcoming AL experiment.

<u>1476</u> Lines	Single Folds			Voting Result			Improvement			
	Base	LH	Mix	NoP	LH	Mix	NL	NM	LM	BM
<b>60</b>	8.12	5.58	4.96	4.72	4.10	2.79	13%	41%	32%	66%
<b>100</b>	6.82	3.99	3.92	3.49	2.67	<u>2.23</u>	23%	36%	16%	67%
<b>150</b>	4.10	3.15	3.03	2.47	2.14	1.66	13%	33%	22%	60%
<b>250</b>	3.24	2.40	2.37	1.70	1.63	<u>1.47</u>	4%	14%	10%	55%
<b>500</b>	2.11	1.73	1.75	1.17	1.13	1.03	3%	12%	9%	51%
<b>1000</b>	1.55	1.30	1.22	0.97	0.88	0.75	9%	23%	15%	52%
<u>1488</u> Lines	Single Folds			Voting Result			Improvement			
	Base	LH	Mix	NoP	LH	Mix	NL	NM	LM	BM
<b>60</b>	7.28	3.97	4.28	4.38	3.05	2.50	30%	43%	18%	66%
<b>100</b>	4.19	2.85	3.20	2.73	2.06	<u>1.84</u>	25%	33%	11%	56%
<b>150</b>	2.96	2.26	2.33	1.81	1.51	1.24	17%	31%	18%	58%
<b>250</b>	2.59	1.82	1.89	1.29	1.21	<u>1.07</u>	6%	17%	12%	59%
<b>500</b>	1.50	1.40	1.38	0.91	0.95	0.79	-4%	13%	17%	47%
<b>1000</b>	1.17	1.06	1.13	0.71	0.72	0.61	-1%	14%	15%	48%
<u>1505</u> Lines	Single Folds			Voting Result			Improvement			
	Base	LH	Mix	NoP	LH	Mix	NL	NM	LM	BM
<b>60</b>	6.54	5.00	5.27	4.58	3.70	3.45	19%	25%	7%	47%
<b>100</b>	4.54	3.96	4.15	3.16	2.82	<u>2.68</u>	11%	15%	5%	41%
<b>150</b>	3.54	3.16	3.16	2.34	2.27	<u>2.02</u>	3%	14%	11%	43%
<b>250</b>	2.85	2.66	2.18	1.98	1.77	<u>1.60</u>	11%	19%	10%	44%
<b>500</b>	2.24	2.18	2.11	1.59	1.60	1.43	-1%	10%	11%	36%
<b>1000</b>	1.84	1.85	1.82	1.35	1.40	1.26	-4%	7%	10%	32%
<u>All</u> Lines	Single Folds			Voting Result			Improvement			
	Base	LH	Mix	NoP	LH	Mix	NL	NM	LM	BM
<b>60</b>	7.98	4.42	4.49	5.34	3.22	2.64	40%	51%	18%	67%
<b>100</b>	4.97	3.38	3.49	2.97	2.41	2.12	19%	29%	12%	57%
<b>150</b>	3.46	2.78	2.87	2.15	1.96	1.67	9%	23%	15%	52%
<b>250</b>	3.06	2.28	2.20	1.79	1.59	1.42	11%	21%	11%	54%
<b>500</b>	2.05	1.86	1.77	1.34	1.27	1.12	5%	16%	12%	45%
<b>1000</b>	1.59	1.46	1.42	1.08	1.07	0.93	2%	14%	13%	42%
<b>Avg.</b>	3.85	2.70	2.69	2.45	1.92	1.65	14%	26%	13%	53%



**Figure 3:** Influence of the number of GT training lines compared for the approaches *Baseline* (no pretraining, no voting), *NoP Voted* (no pretraining, voted), and *Mix Voted* (pretrained with different mixed models, voted) on a logarithmic scale for CER.

leading to an additional reduction of recognition errors by over 13% without an apparent correlation regarding the amount of training GT.

Comparing the best method (Mix + Voting) with the baseline, i.e. the default OCRopus approach (training a single model without any pretraining or voting), shows the superiority of the proposed approach yet more clearly. Even the error rates of strong individual models trained on a 1,000 lines of GT are reduced by more than 40% on average. In general, a substantial amount of GT (>250 lines) is required in the standard OCRopus training (see ‘baseline’ in Figure 2) to match the result achieved by a mere 60 lines when incorporating mixed pretraining and voting, indicating a GT saving factor of 4 or more. A similar factor can be derived when considering the average baseline CER for 1,000 lines of GT.

**Table 3:** Results of comparing active to passive learning. *Base Lines* is the number of lines used to train the voters of the previous iteration. The lines added (*Add. Lines*) randomly (*RDM*) or by maximizing the  $LDR_{\emptyset}$  (*AL*) correspond to 50% of the base lines. Compared are the resulting error rates (*CER*) after performing a confidence vote and (in case of *RDM*) an averaging calculation. Finally, (*Avg. Gain*) shows the average improvement of the voters trained by *AL*.

Book	Base		Add. Lines Rdm/AL	CER		Average Gain
	Lines	CER		Rdm	AL	
1476	100	2.23	50	1.80	1.31	26%
	250	1.47	125	1.18	0.90	24%
1488	100	1.84	50	1.54	1.05	32%
	250	1.07	125	0.86	0.65	24%
1505	100	2.68	50	2.17	2.21	-2%
	250	1.81	125	1.60	1.57	2%

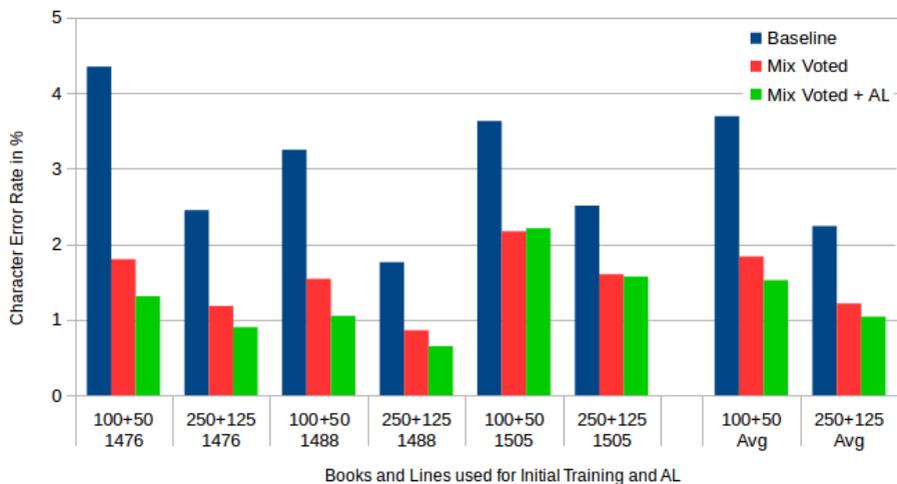
## 4.2 Incorporating Active Learning

To select additional training lines we utilized the models (voters) obtained from the previous experiment. Each model recognized the GT lines set aside for *AL* and the best candidates were determined by choosing the ones with the highest  $LDR_{\emptyset}$  as explained in section 3.5. Next, the candidates underwent a quick visual inspection in order to sort out lines where a positive impact on the training was considered highly unlikely due to very rare segmentation errors or extreme degradations. Indeed, this adds a bit of subjectivity to the task but we expected the decision whether to keep or drop a line to be trivial in most cases. However, in our experiments we never skipped a line proposed by the *AL* approach despite coming across several borderline cases which will be discussed below.

We performed two experiments starting with different numbers of (base) GT lines (100/250) which we kept from the previous experiment. For the passive learning approach we added an additional 50% (50/125) of randomly selected lines. This was performed five times and the results were averaged. As for *AL* we chose the lines by following the principle of maximal disagreement incorporating the  $LDR_{\emptyset}$ . Since the number of characters per line may vary we made an effort to select only as many lines as necessary to match the average amount of characters in the passive learning approach.

After selecting the lines the base fold setup was kept and we distributed the additional GT evenly over the five folds to ensure an effect on the training itself but also on the selection of the best model. Afterwards, the training was started from scratch/from the default mixed models while the voters were discarded.

Since the previous experiment showed the superiority of the mixed pretraining approach we decided to omit the pretraining using *LH* during the upcoming experiments. Table 3 shows the results.



**Figure 4:** Results of the AL experiments for three books at two different sets of lines comparing the *Baseline* (no pretraining, no voting), *Mix Voted* (pretrained with different mixed models, voted) and *Mix Voted + AL* (Mix with additional lines chosen by AL).

Incorporating AL leads to lower CERs for four out of six tested scenarios. While important improvements with an average gain of almost 27% can be reported for 1476 and 1488, 1505 does not improve at all (see the discussion in the next section).

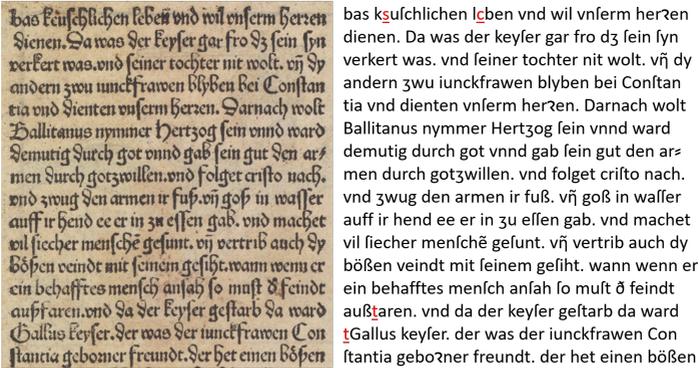
Moreover, it is worth mentioning that no clear influence of the number of the GT lines available for training can be inferred on the basis of these results. Even when starting from an already quite comprehensive GT pool of 250 lines AL yielded an average gain of 16% compared to randomly chosen lines.

Finally, Figure 4 sums up the results by comparing the baseline to the best pretraining (Mix) approach combined with confidence voting with and without AL.

## 5 Discussion

The experiments show that the combination of pretrained models and confidence voting is an effective way to further improve the achievable CER on early printed books. While the obtainable gain is highest when the number of available GT lines is small, a substantial reduction of OCR errors can still be expected even when training with several hundreds of lines.

An interesting result of our experiments is that the variability of the voters has a clear influence on the voting result and can even outweigh a superior individual quality of the single voters. This explains why using a variety of models for pretraining considerably



**Figure 5:** Snippet of a scanned page for 1488 and its best OCR output by combining mixed pretraining and voting. The resulting CER of 0.60% was achieved by training on 1,000 GT lines. The remaining four recognition errors are marked in red and underlined.

outperformed  $N$ -fold training from the LH model even though it represented the best fitting one of the available mixed models. Since training from an available model skips the random initialization of values in the weight matrix for pre-existing characters an important chance of introducing diversity to the training process is skipped, resulting in quite similar models even when trained on different but still heavily overlapping folds of training GT.

Following the proposed approach of combining a mixed pretraining with confidence voting allows for a substantially more efficient use of the available GT. On average we were able to achieve the same results as the standard OCRopus approach requiring less than one fourth of the number of GT lines to do so. Moreover, a tiny amount of GT – only 60 lines – was enough to reach an average CAR of close to 97.5%. Only two out of six books showed a CER greater than 3% but comfortably surpassed this value when adding another 40 lines of GT, raising the average to almost 98% character accuracy, which is already considered good enough for many areas of application.

Despite these improvements, there are opportunities for optimizing the achieved results even further, e.g. in applications where the goal is to manually check and correct an entire book in order to obtain a CER of close to 0%. The experiments showed that our method also significantly outperforms the standard approach when training on a very comprehensive GT pool of 1,000 lines, resulting in an average CER of less than 1%. As an example, Figure 5 gives an impression of input (scanned page image) and output (best possible OCR result) for the 1488 printing with a CER of 0.60% reached by training on 1,000 lines combining pretraining using a variety of mixed models and confidence voting.

Even if transcription of an entire book is intended, the goal still should be to minimize the CER by investing the least possible manual correction effort. Therefore, an iterative training approach makes sense and a efficient selection of further training lines is important. Our experiments on AL showed that choosing additional lines in an informed manner can offer an even more efficient way to use the available GT. Despite one of the books not responding at all to the proposed method due to heavily and inconsistently degraded glyphs the three evaluated books still showed an improvement of 17% compared to the mixed voting approach when selecting additional training lines randomly for transcription. It is worth mentioning that during our experiments the number of lines presented to the committee of voters was considerably smaller than in a real world application scenario where it might be sensible to take all yet unknown lines into consideration in order to choose the most promising ones in terms of information gain. This might have a positive effect on the achievable results of the AL approach.

A likely explanation for the lack of improvements of the 1505 book by AL is the degree and type of degradations of the lines queried for training by the voting committee which is illustrated by some example lines shown in Figure 6. The lines on the left are examples the voters fully agreed on and, as expected, got recognized correctly by the base models trained with 100 lines of GT. On the right some of the lines for which the committee disagreed the most are shown, i.e. the ones with highest  $LDR_{\emptyset}$ . For 1476 (top) the lines shown had a ratio of 0.45 and CER of 9.44%. There are some signs of degradation, mostly moderately faded glyphs. The worst lines selected from 1488 (middle, 0.72  $LDR_{\emptyset}$ , 33.68% CER) mostly suffered from noise while the glyphs of 1505 (bottom, 0.64  $LDR_{\emptyset}$ , 30.77% CER) frequently show severe deformations. This might be a sensible explanation why AL works very well for 1476 and 1488 but not at all for 1505. Despite the fading and the noise the glyphs of 1476 and 1488 look much more regular than the deformed ones of 1505, at least to the human eye. Therefore, the models trained for 1476 and 1488 using AL learned to see through the effects of fading and noise and earned additional robustness, resulting in a considerable gain in CER. In the case of 1505 the AL models were fed many lines showing severe but very irregular degradations which may have led to an increased robustness but probably did not improve the recognition capability of regular lines as much as the passive learning lines did.

## 6 Conclusion and Future Work

In this paper we proposed a combination of pretraining, confidence voting, and AL in order to significantly improve the achievable CER on early printed books. The methods were shown to be very effective for different amounts of GT lines typically available from several hours' transcription work of early printings.

In the future we aim to utilize the positive effect of more diverse voters even further. In general, a higher degree of diversity can be achieved in two ways:

<b>Dat dye nuyffer tsamen moedy ch</b>	<b>W guedt ind oick v lijff</b>
<b>Der nuyffer starcke moelen torn</b>	<b>freuwoy ch rieffē sy stait vast yr fromē ritter ind</b>
<b>Garmhartzich gnedige frauw</b>	<b>Zo mynrebwedēz ind do con</b>
<b>fagen ir solt sye abgötter anbeten Des will er</b>	<b>zu nam.</b>
<b>vnd bereyter sy schar wol nach weyßheyt. vij</b>	<b>gen freuden.</b>
<b>vmb Seinen Dienst Den du mir gethan hast. vij</b>	<b>Sein boten gar fer. vnd embot sant Anton hin</b>
<b>te aut inopia: z in sua versutia z opibus</b>	<b>med s tenebris. Prouer. xx.</b>
<b>fensozē fugiet colon? .i. agricola sine vi. i.</b>	<b>vet mel. quinto.</b>
<b>mutabile ac tranſitorioiū se cōuertunt:</b>	<b>curare mortalia aut nullā vindictā facturū</b>

Figure 6: Example lines of the three books 1476 (top), 1488 (middle), and 1505 (bottom) which were presented to the committee. The left column shows perfectly recognized lines ( $LDR_{\emptyset}$  of 0.0). On the right some of the most erroneous lines are presented ( $LDR_{\emptyset}$  of 0.45 (1476), 0.72 (1488), and 0.64 (1505)).

1. By the inclusion of more mixed models for pretraining. Since there are only a few good mixed models freely available to date, there is a great need and sharing is key. To lead by example we made part of our GT data and models available online<sup>12</sup>.
2. By varying the network structure used for training representing a viable leverage point to increase diversity even further. First steps in this direction have recently been made by Breuel (2017) and Wick et al. (*Comparison of OCR Accuracy on Early Printed Books using the Open Source Engines Calamari and OCRopus*; this issue).

Clearly, the combination of both intended improvements also represents a very promising approach.

To optimize the achievable results, extensive experiments regarding parameter optimization are required. This includes the number of folds/voters, the kind of network they have been trained on, and the method of combination of mixed models used for pretraining, as well as the number of lines the training is performed on.

As for AL an important additional approach is to not only utilize it in order to choose new training lines before the actual training process but also to get involved during the training itself. The standard OCRopus approach is to randomly select lines and feed them to the network. A more efficient method might be to decrease the chance to get picked for lines which already got perfectly recognized and consequently increase it for lines which still cause the current model a lot of problems.

<sup>12</sup>[https://github.com/chreul/OCR\\_Testdata\\_EarlyPrintedBooks](https://github.com/chreul/OCR_Testdata_EarlyPrintedBooks)

Concerning the maximal disagreement approach for line selection, it would be interesting to experiment with other ways to determine those lines that offer a maximal information gain. Utilizing the intrinsic OCRopus confidence values that we already use during our voting approach comes to mind but also measures like the Kullback-Leibler-Divergence.

Finally, despite our focus on early printed books the proposed methods are applicable to newer works as well. Especially 19th century Fraktur presents an interesting area of application. Since the typically used Fraktur typesets are more regular than those of the books used during our experiments the goal is to produce a mixed model with excellent predictive power and to avoid book specific training at all.

Nevertheless, the present paper is just a first step to the larger goal of creating an effective, open-source, computer-assisted OCR workflow that is automated to the largest extent possible. In the context of the OCR-D project this also encompasses methods of page segmentation (Reul et al., 2017b) in the preprocessing phase as well as postcorrection (Fink et al., 2017) in the postprocessing stage which are just as important for the quality of the end result as the recognition process itself. To enable further progress in the direction of better recognition results and better polyfont models, we made an extensive set of historical GT data comprising over 300,000 lines available in the GT4HistOCR (*Ground Truth for Historical OCR*) corpus under a CC-BY-SA-4.0 license at Zenodo<sup>13</sup> (see the article of Springmann, Reul, Dipper, Baiter in this issue).

## References

- Al Azawi, M., Liwicki, M., and Breuel, T. M. (2015). Combination of multiple aligned recognition outputs using WFST and LSTM. In *Document Analysis and Recognition (ICDAR), 2015 13th Int. Conf. on*, pages 31–35. IEEE.
- Boschetti, F., Romanello, M., Babeu, A., Bamman, D., and Crane, G. (2009). Improving OCR accuracy for classical critical editions. *Research and Advanced Technology for Digital Libraries*, pages 156–167.
- Breuel, T. M. (2017). High performance text recognition using a hybrid convolutional-LSTM implementation. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 11–16. IEEE.
- Breuel, T. M., Ul-Hasan, A., Al-Azawi, M. A., and Shafait, F. (2013). High-performance OCR for printed English and Fraktur using LSTM networks. *12th International Conference on Document Analysis and Recognition*, pages 683–687.
- Fink, F., Schulz, K. U., and Springmann, U. (2017). Profiling of OCR’ed historical texts revisited. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*, DATeCH2017, pages 61–66, New York, NY, USA. ACM.
- Fischer, A., Wüthrich, M., Liwicki, M., Frinken, V., Bunke, H., Viehhauser, G., and Stolz, M. (2009). Automatic transcription of handwritten medieval documents. In *15th International Conference on Virtual Systems and Multimedia, 2009 (VSM’09)*, pages 137–142. IEEE.

<sup>13</sup><https://zenodo.org/record/1344131>

- Handley, J. C. (1998). Improving OCR accuracy through combination: A survey. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE Int. Conf. on*, volume 5, pages 4330–4333. IEEE.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Kirchner, F., Dittrich, M., Beckenbauer, P., and Nöth, M. (2016). OCR bei Inkunabeln – Offizinspezifischer Ansatz der UB Würzburg. *ABI Technik*, 36 (3):178–188.
- Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pages 231–238.
- Liwicki, M., Bunke, H., Pittman, J. A., and Knerr, S. (2011). Combining diverse systems for handwritten text line recognition. *Machine Vision and Applications*, 22(1):39–51.
- Lopresti, D. and Zhou, J. (1997). Using consensus sequence voting to correct OCR errors. *Computer Vision and Image Understanding*, 67(1):39–47.
- Lund, W. B., Walker, D. D., and Ringger, E. K. (2011). Progressive alignment and discriminative error correction for multiple OCR engines. In *Document Analysis and Recognition (ICDAR), 2011 Int. Conf. on*, pages 764–768. IEEE.
- Nartker, T. A., Rice, S. V., and Lumos, S. E. (2005). Software tools and test data for research and testing of page-reading OCR systems. In *Electronic Imaging 2005*, pages 37–47. International Society for Optics and Photonics.
- Reul, C., Dittrich, M., and Gruner, M. (2017a). Case study of a highly automated layout analysis and OCR of an incunabulum: 'Der Heiligen Leben' (1488). In *Proceedings of the 2Nd Int. Conf. on Digital Access to Textual Cultural Heritage, DATeCH2017*, pages 155–160, New York, NY, USA. ACM.
- Reul, C., Springmann, U., and Puppe, F. (2017b). Larex: A semi-automatic open-source tool for layout analysis and region extraction on early printed books. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage, DATeCH2017*, pages 137–142, New York, NY, USA. ACM.
- Reul, C., Springmann, U., Wick, C., and Puppe, F. (2018). Improving OCR accuracy on early printed books by utilizing cross fold training and voting. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 423–428. IEEE.
- Reul, C., Wick, C., Springmann, U., and Puppe, F. (2017c). Transfer learning for OCRopus model training on early printed books. *027.7 Journal for Library Culture*, 5(1):38–51.
- Rice, S. V., Jenkins, F. R., and Nartker, T. A. (1996). *The fifth annual test of OCR accuracy*. Information Science Research Institute.
- Rice, S. V., Kanai, J., and Nartker, T. A. (1992). *A report on the accuracy of OCR devices*. University of Nevada, Information Science Research Institute.
- Rice, S. V., Kanai, J., and Nartker, T. A. (1994). An algorithm for matching OCR-generated text strings. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(05):1259–1268.
- Rice, S. V. and Nartker, T. A. (1996). The ISRI analytic tools for OCR evaluation. *UNLV/Information Science Research Institute, TR-96-02*.

- Rydberg-Cox, J. A. (2009). Digitizing latin incunabula: Challenges, methods, and possibilities. *Digital Humanities Quarterly*, 3(1).
- Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.
- Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM.
- Springmann, U. (2015). Ocrocis: A high accuracy OCR method to convert early printings into digital text – A Tutorial. <http://cistern.cis.lmu.de/ocrocis/tutorial.pdf>.
- Springmann, U. and Fink, F. (2016). CIS OCR Workshop v1.0: OCR and postcorrection of early printings for digital humanities.
- Springmann, U., Fink, F., and Schulz, K. U. (2016). Automatic quality evaluation and (semi-) automatic improvement of mixed models for OCR on historical documents. *CoRR*.
- Springmann, U. and Lüdeling, A. (2017). OCR of historical printings with an application to building diachronic corpora: A case study using the RIDGES herbal corpus. *Digital Humanities Quarterly*, 11(2).
- Springmann, U., Najock, D., Morgenroth, H., Schmid, H., Gotscharek, A., and Fink, F. (2014). OCR of historical printings of Latin texts: problems, prospects, progress. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, DATeCH '14*, pages 57–61, New York, NY, USA. ACM.
- Stäcker, T. (2014). Konversion des kulturellen Erbes für die Forschung: Volltextbeschaffung und -bereitstellung als Aufgabe der Bibliotheken. *o-bib. Das offene Bibliotheksjournal*, 1(1):220–237.
- Ul-Hasan, A. and Breuel, T. M. (2013). Can we build language-independent OCR using LSTM networks? *Proceedings of the 4th International Workshop on Multilingual OCR*, page 9.
- Wemhoener, D., Yalniz, I. Z., and Manmatha, R. (2013). Creating an improved version using noisy OCR from multiple editions. In *Document Analysis and Recognition (ICDAR), 2013 12th Int. Conf. on*, pages 160–164. IEEE.
- Wick, C. and Puppe, F. (2017). Leaf identification using a deep convolutional neural network. *arXiv preprint arXiv:1712.00967*.
- Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.