New Approaches to OCR for Early Printed Books

«Dig*Italia*» 2-2020
DOI: 10.36181/digitalia-00015

Nikolaus Weichselbaumer - *University of Mainz*
Mathias Seuret - *University of Erlangen-Nuremberg*
Saskia Limbach - *University of Milan*
Rui Dong - *Northeastern University*
Manuel Burghardt - *Leipzig University*
Vincent Christlein - *University of Erlangen-Nuremberg*

Books printed before 1800 present major problems for OCR. One of the main obstacles is the lack of diversity of historical fonts in training data. The OCR-D project, consisting of book historians and computer scientists, aims to address this deficiency by focussing on three major issues. Our first target was to create a tool that identifies font groups automatically in images of historical documents. We concentrated on Gothic font groups that were commonly used in German texts printed in the 15th and 16th century: the well-known Fraktur and the lesser known Bastarda, Rotunda, Textura und Schwabacher. The tool was trained with 35,000 images and reaches an accuracy level of 98%. It can not only differentiate between the above-mentioned font groups but also Hebrew, Greek, Antiqua and Italic. It can also identify woodcut images and irrelevant data (book covers, empty pages, etc.). In a second step, we created an online training infrastructure (okralact), which allows for the use of various open source OCR engines such as Tesseract, OCRopus, Kraken and Calamari. At the same time, it facilitates training for specific models of font groups. The high accuracy of the recognition tool paves the way for the unprecedented opportunity to differentiate between the fonts used by individual printers. With more training data and further adjustments, the tool could help to fill a major gap in historical research.

OCR-D

In past decades, many libraries in Germany have started to digitise their holdings. The numerous digital copies of early printed books have been linked to the matching records in the respective national bibliographies, VD16, VD17 and VD18[1]. Taken together, there are now hundreds of thousands of links to digital copies. This is already a significant achievement; yet it would mean a huge step forward for scholarship if these items were available for full-text searches and further processing. Therefore the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) organised a workshop in 2014 in

[1] <http://www.vd16.de/>; <http://www.vd17.de/>; <http://www.vd18.de/>.

which experts assessed how new developments in Optical Character Recognition (OCR) can be used to achieve this goal[2]. OCR is a complex process that comprises more than the character recognition itself. It involves pre-processing (e.g. image de-noising, binarisation, etc.); layout analysis (recognising elements such as headings and illustrations); and post-processing (correcting errors).

The OCR-D project aims to create a conceptual and technical framework that allows for full text transformation of any digital copy in VD16, VD17 and VD18[3]. For this purpose, the individual steps of automatic text recognition are broken down, allowing users to trace them in the open-source software of OCR-D[4]. Users can then adapt the workflow to their specific needs, e.g. when the layout of particular early printed books need specific settings.

The OCR-D project was coordinated by four partners: the Berlin-Brandenburg Academy of Sciences and Humanities (BBAW), the Herzog-August Library Wolfenbüttel (HAB), the Berlin State Library (SBB) and the Karlsruhe Institute of Technology (KIT)[5]. In the first phase, the project identified major challenges for early printed books and funded eight module projects to address these issues.

Recognising Font Groups

Our module project addresses specific aspects of print in early modern Europe[6].
One of the biggest obstacles for OCR use in early printed books is the fact that OCR engines are usually trained with modern-day fonts. This ignores the large regional and stylistic variety in fonts used in printed texts before 1800 and leads to high error rates in text recognition. Our premise was the following: better OCR results can be created by being able to recognise automatically the font group used in an early printed book which would allow users to choose the best OCR-model for each text; in a second step users would also be able to train their own font-specific model for particularly difficult fonts (such as one of the fonts used by William Caxton, the first printer in England, see below).

[2] https://www.dfg.de/en/index.jsp.

[3] Clemens Neudecker - Konstantin Baierer - Maria Federbusch - Matthias Boenig - Kay-Michael Würzner - Volker Hartmann - Elisa Herrmann, *OCR-D: An end-to-end open source OCR framework for historical printed documents*, in: *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, 2019, p. 53–58, <https://ocr-d.de/en/>.

[4] Since early 2020, a prototype of the OCR-D software is used in nine libraries, see: Konstantin Baierer - Matthias Boenig - Elisabeth Engl - Volker Hartmann - Clemens Neudecker - Reinhard Altenhöner - Alexander Geyken - Johannes Mangei - Rainer Stotzka, *OCR-D kompakt: Ergebnisse und Stand der Forschung in der Förderinitiative*, «BIBLIOTHEK–Forschung und Praxis» 44 (2020), n. 2, p. 1–13.

[5] <https://www.bbaw.de/en/>; <http://www.hab.de/en/home.html>; <https://staatsbibliothek-berlin.de/en/>; <https://www.kit.edu/english/index.php>.

[6] https://www.buchwissenschaft.uni-mainz.de/forschung/modellrepositorium-ocr-d/#englisch.

To this end, the project had three major objectives:

- train a neural network to recognise font groups commonly used in early modern books;
- build infrastructure to allow for easy training of all important open source OCR-engines;
- making our software publicly available for free.

Training a neural network requires a large amount of data. In our case, we needed images of book pages for which we knew which font group they showed. Thankfully, the TW (Typenrepertorium der Wiegendrucke) offered a good point of departure[7]. The TW has been developed over more than a century, listing almost every font ever used in incunabula (books printed between 1450 and 1500). This comprehensiveness made it ideal for our project. However, one has to keep in mind that the data in the TW is not complete. Most books were printed with two (or more) fonts, yet often the TW has only identified one of them. Moreover, it is not indicated in the records where specifically the font is used in the book, e.g. only on the title page or as a running header.

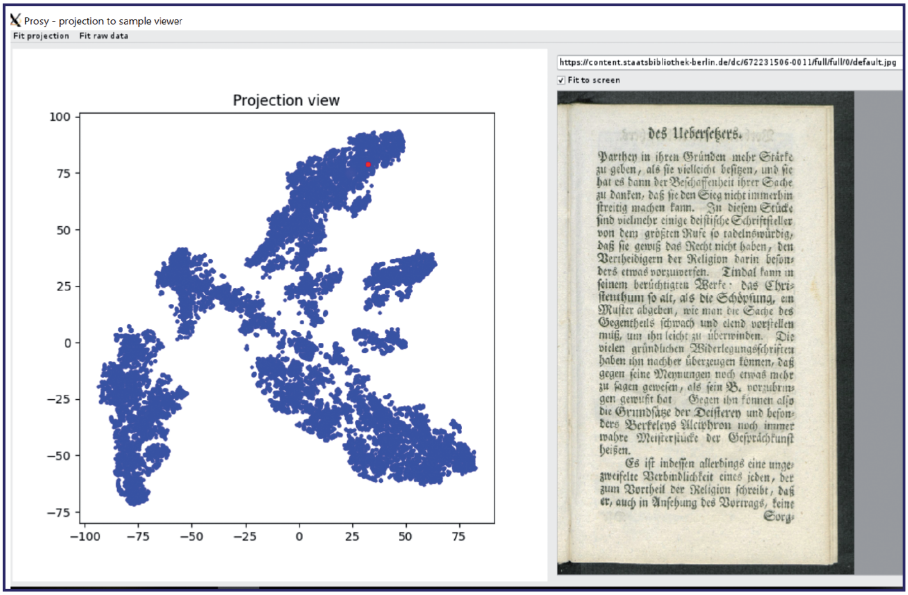
Such pages with multiple font groups created much noise for the network, which prevented it from learning accurately. Thus, it was necessary for a human expert to go through the images first and indicate manually which font was used on which page. For this, we made use of a very comprehensive list of fonts that Dr. Oliver Duntze from the TW project kindly shared with us. The list differentiates between the many Gothic fonts and Antiqua fonts and groups them together. We could then specifically look for certain font groups and selected images for them. On top of that, we accumulated as much data as possible from our cooperating libraries[8].

We divided about 6,000 fonts of the TW into 6 groups: Textura, Rotunda, Bastarda, Schwabacher, Gotico-Antiqua and Antiqua. We then accumulated material for two additional font groups often found in early printed books: Greek and Hebrew. Finally we also included fonts which came into use early in the 16th century: Fraktur and Italic.

[7] https://tw.staatsbibliothek-berlin.de/.

[8] We were graciously supported by a number of libraries which provided us with data: the State Library in Berlin, the University Library of Cologne, the University Library of Erlangen, the State and University Library Göttingen, the University Library of Heidelberg, the State Library Munich, the State Library Stuttgart, the Herzog August Bibliothek Wolfenbüttel. We estimate that we were given access to more than 250 million digitised pages. We were especially thankful for the support of the British Library (London) which allowed us to work with their digitised incunabula collection. The British Library houses one of the largest and most diverse collections of books printed in the 15th century and the institution recently scanned their entire collection with outstanding image quality. Although the images are not online yet, the library sent us the data on a hard drive and allowed us to use the images for our research purposes.

After a first test, which showed promising but improvable results, we attempted to better understand the usefulness of these groups from a computer's point of view[9]. For this, we first trained a neural network on several thousand pages. Then, we randomly selected 18,000 unlabelled page images from the holdings of the Berlin State Library and extracted feature vectors for each of them. To do so, we applied the network on the whole surface of each page, and stored the 384-dimensional output of the penultimate layer of the network – a typical feature extraction approach. As such high-dimensional data cannot be directly displayed, nor be understood by a human, we projected it into a 2-D space using the t-distributed stochastic neighbor embedding (t-SNE)[10]. Each of the 18,000 images is displayed as a blue dot (see Fig. 1). By clicking on any of the dots (which then turns red), the corresponding page is displayed on the right side.



https://content.staatsbibliothek-berlin.de/dc/672231506-0011/full/full/0/default.jpg

☑ Fit to screen

des Ueberſetzers.

Parthey in ihren Gründen mehr Stärke zu geben, als ſie vielleicht beſitzen, und ſie hat es dann der Beſchaffenheit ihrer Sache zu danken, daß ſie den Sieg nicht immerhin ſtreitig machen kann. In dieſem Stücke ſind vielmehr einige deiſtiſche Schriftſteller von dem größten Rufe ſo tadelnswürdig, daß ſie gewiß das Recht nicht haben, den Vertheidigern der Religion darin beſonders etwas vorzuwerfen. Tindal kann in ſeinem berüchtigten Werke: das Chriſtenthum ſo alt, als die Schöpfung, ein Muſter abgeben, wie man die Sache des Gegentheils ſchwach und elend vorſtellen muß, um ihn leicht zu überwinden. Die vielen gründlichen Widerlegungsſchriften haben ihn nachher überzeugen können, daß gegen ſeine Meynungen noch etwas mehr zu ſagen geweſen, als ſein B. vorzubringen gewußt hat Gegen ihn können alſo die Grundſätze der Deiſterey und beſonders Berkeleys Alciphron noch immer wahre Meiſterſtücke der Geſprächkunſt heißen.

Es iſt indeſſen allerdings eine ungezweifelte Verbindlichkeit eines jeden, der zum Vortheil der Religion ſchreibt, daß er, auch in Anſehung des Vortrags, keine Sorg-

Figure 1. *Clustering of 18k images from books in the Berlin State Library. The shown book page is located in the top part of the upper cluster (red dot). Screenshot of DICI, the tool used to check the test results. <https://github.com/seuretm/dici>*

Once we had all individual pages displayed as blue dots, a human expert carefully checked thousands of pages for common features. It turned out that the clusters indeed matched some font groups, such as Antiqua, Rotunda, Textura, Bastarda and Fraktur (as seen in Fig. 2).

[9] Nikolaus Weichselbaumer - Mathias Seuret - Saskia Limbach - Vincent Christlein - Andreas Maier, *Automatic Font Group Recognition in Early Printed Books*, in: *Digital Humanities im deutschsprachigen Raum (DHd) 2019. 6th International Conference 25-29 March 2019, Universitäten zu Mainz und Frankfurt*, p. 84-87, <https://doi.org/10.5281/zenodo.2596095>.

[10] Laurens van der Maaten and Geoffrey Hinton, *Visualizing data using t-SNE*, Journal of Machine Learning Research 9 (2008), p. 2579-2605.

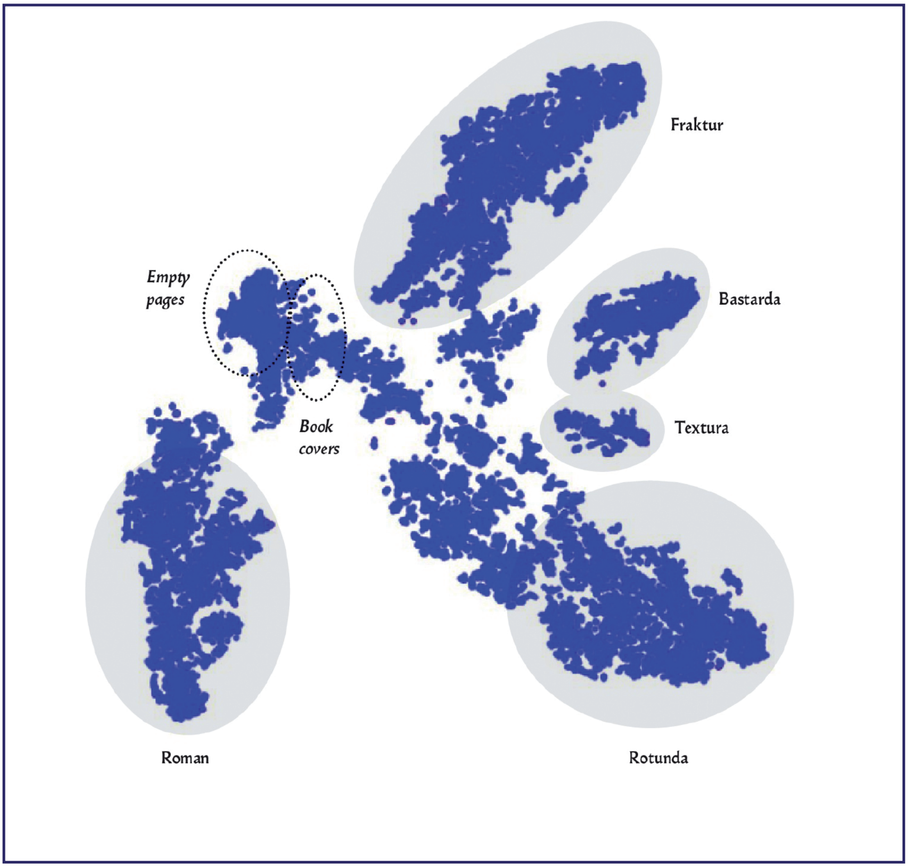


Figure 2. *Results in t-sne, showing that certain font groups were clustered together*

Other font groups were more difficult to find. Italic pages, which are not marked in Fig. 2, mostly appeared on top of the Antiqua cluster, along with many pages containing a mixture of Antiqua and Italic. Mixing these two font groups was quite common in the early modern period. Hence, there is no clear Italic cluster.

The results also showed that it was important to create more groups, which allowed us to teach the network to sort out "distractions", such as empty pages or book covers. The network grouped them together already. However, if the network is not taught specifically that these images are different, i.e. do not contain a font group, it will attribute them to one of the selected font groups. Therefore, we created two additional groups. On the one hand, the group "other font" includes handwritten notes and non-Latin alphabets beyond Greek and Hebrew (such as Arabic and Farsi). On the other hand, we created the group "not a font" which includes blank pages, book covers, woodcuts, etchings, paintings, printer marks, ornaments and remnants of the digitisation process, such as colour charts, scanner beds and accidentally reproduced hands.

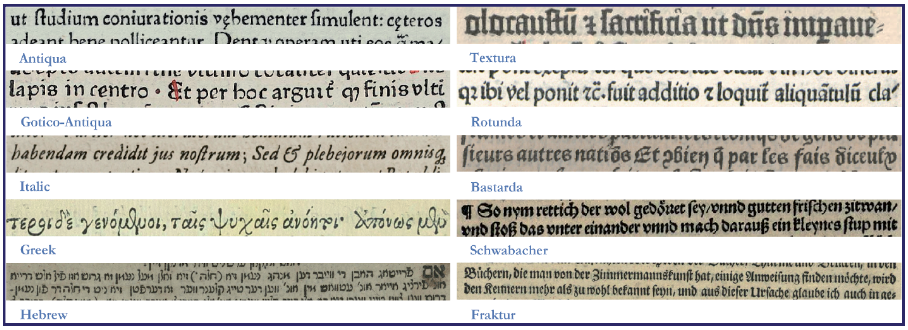

Antiqua
Textura
Gotico-Antiqua
Rotunda
Italic
Bastarda
Greek
Schwabacher
Hebrew
Fraktur

Figure 3. *Samples of the font groups used in the project*

Our next step was to prepare a large amount of data for training a deep neural network to identify font groups, since no such dataset existed before (see Fig. 3). We built it in an iterative way: we used information from the TW to identify incunabula containing the desired font groups and collected roughly 100 pages for each group. We used these pages for training a relatively small network, inspired by existing script identification methods[11]. This network helped us to identify useful pages without any distractions out of the data provided by our partner libraries. This approach was significantly more efficient than a manual search.

We labelled the selected pages using two separate approaches: first, we used labelbox and uploaded a selection of images from our partner libraries[12]. Then, we asked a number of participants to assign the correct label(s) and specify the main font if there was more than one. Second, we used the rest of the material our partners supplied to select pages manually with only a single font group. For this we prepared a batch of images from multiple libraries in one folder and extracted the "not a font" images in thumbnail view. We examined the remaining images and deleted every page with more than one font group. Finally, we transferred each image in the right single class folder. This second method proved to be much more efficient, since we could reduce loading times considerably and got significantly more data for underrepresented font groups, such as Textura and Rotunda. As a result our dataset consists mostly of pages with only one font group per page. We attempted to include pages from all editions of Gotico-Antiqua that were digitally available under free licenses in order to compensate for the form variety within this font group. This dataset is now freely available

[11] Florence Cloppet - Véronique Églin - Van Cuong Kieu - Dominique Stutzmann - Nicole Vincent, *ICFHR2016 Competition on the Classification of Medieval Handwritings in Latin Script,* in: *ICFHR, 2016*, p. 590–595; Florence Cloppet - Véronique Églin - Marlène Helias-Baron - Van Cuong Kieu - Nicole Vincent - Dominique Stutzmann, I*CDAR2017 Competition on the Classification of Medieval Handwritings in Latin Script*, in *ICDAR, 2017*, p. 1371–1376.

[12] https://labelbox.com.

online[13]. While our dataset was growing, we tried various standard and specialised neural network architectures. We started with residual networks (ResNet) with 18 and 50 layers, greatly decreasing the amount of neurons in the various layers to prevent over-fitting[14]. We also tried VGG-16, and combinations of variational autoencoders and classifiers[15]. We found out that for small amounts of data, the design of architectures specific for this task tends to greatly improve the results. However, with the amount of data we labelled, differences between architectures almost vanished, and we settled on a DenseNet-121[16].

A challenge, which is not entirely solved yet, is the case of pages with multiple font groups. Our labelled data contains many such pages, but processing them correctly with the network is not straightforward. The network attributes a "score" to each of its possible outputs. This score is an unbounded real number, and we decided that the highest value corresponds to the answer of the network. If several font groups are present on a page, then only one will be detected – typically the one covering the largest surface, as this leads to a higher score. Using a global score threshold from which a font group is detected is not directly feasible, as there is no simple way to distinguish between a misdetection (e.g. a medium score for Antiqua since there are also lines in Italic), and a correct detection of a small surface (e.g. a single word in Antiqua in the middle of a paragraph in Fraktur).

This issue could be tackled in two ways. First, if the training data contained location information for the different font groups, then the network could learn to provide reliable results even for small amounts of text, as localisation information could be used during the training phase. This would, however, require a vast amount of manual labour to prepare. In addition to simply indicating which font groups are present on a page, the experts preparing the data would have to manually draw rectangles or polygons over many words or parts of words in secondary fonts. This time-consuming procedure would have to be performed on thousands of pages.

[13] Mathias Seuret - Saskia Limbach - Nikolaus Weichselbaumer - Andreas Maier - Vincent Christlein, *Dataset of Pages from Early Printed Books with Multiple Font Groups*, in: *HIP '19. Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, p. 1–6, <https://doi.org/10.1145/3352631.3352640>.

[14] Kaiming He - Xiangyu Zhang - Shaoqing Ren - Jian Sun, *Deep residual learning for image recognition. 2016*, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770-778, <https://doi.org/10.1109/CVPR.2016.90>.

[15] Karen Simonyan - Andrew Zisserman, *Very deep convolutional networks for large-scale image recognition*. 2014. *arXiv preprint* <arXiv:1409.1556>; Diederik P. Kingma - Max Welling, *Auto-encoding variational Bayes*. 2013. *Proceedings of the 2nd International Conference on Learning Representations. arXiv preprint* <arXiv:1312.6114>.

[16] Gao Huang - Zhuang Liu - Laurens Van Der Maaten - Kilian Q. Weinberger, *Densely connected convolutional networks,* in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, p. 2261-2269, <https://doi.org/10.1109/CVPR.2017.243>.

The second option we investigated is to command the network to indicate which areas of its input could lead to a specific font group output, and how much[17]. This is illustrated in the two images below, which we took from an incunable currently in the collection of the British Library (see Fig. 4). The intensity of the heatmap (or highlighting) indicates which specific regions of the page get related to a font group by the neural network. Thus, in the case of mixed content, the network is able to provide location information – even if it was never trained for this purpose.

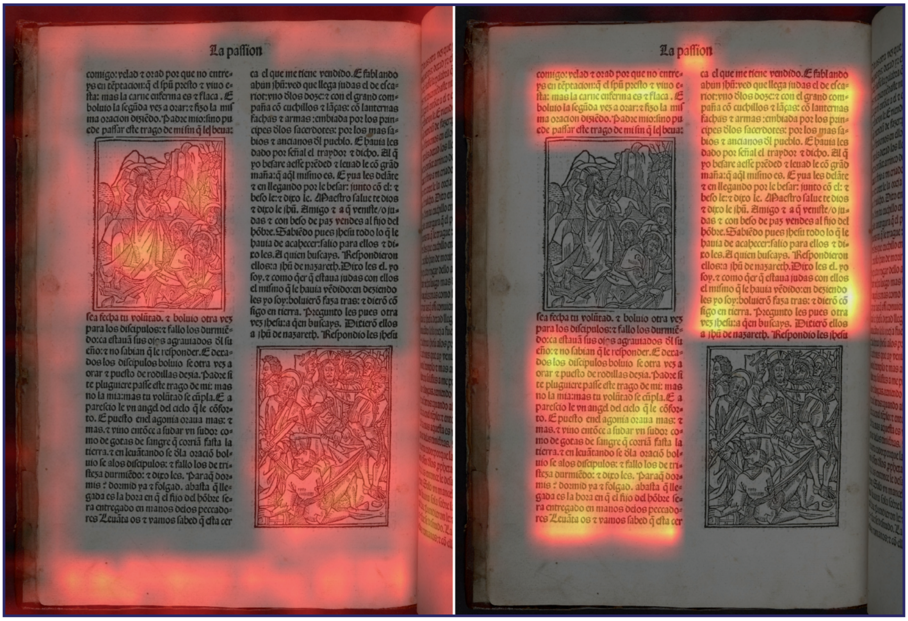


Figure 4. *Heatmaps showing confidences for Not a font (left) and Rotunda (right). Sample page: Jacobus de Voragine, Legenda aurea sanctorum (...), [Burgos: Juan de Burgos, about 1499], London, British Library: IB.53312. (GW M11511), p. 10*

However, generating such heatmaps is time-consuming and therefore not well suited for processing large amounts of images, such as library collections. On top of that, storing the results for later use would require much disk space: when we store only one image plus color codes for labeling the pixels, the results may be difficult to evaluate. It would be better to store each heatmap as a single image. Yet, this takes up a lot of space. Thus, while these results are surprisingly good, we decided not to pursue this approach for our project. After all, our objective was to deal with large amounts of data, and using a slow, computationally expensive and

[17] Using Grad-CAM, a method estimating where the source of a decision taken by a neural network is, see Ramprasaath R. Selvaraju - Michael Cogswell - Abhishek Das - Ramakrishna Vedantam - Devi Parikh - Dhruv Batra, *Grad-cam: Visual explanations from deep networks via gradient-based localization,* in: *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, p. 618-626, <https://doi.org/10.1109/ICCV.2017.74>.

power-consuming method is simply not acceptable, regardless of the accuracy achieved. In addition to these considerations, as the font recognition step can potentially be applied after the layout analysis, the labelling can be done with a finer granularity than page-level.

Training and Using OCR models

In the second part of our project, we created an online training infrastructure (okralact), which makes it much easier to train OCR models for different engines[18]. We focussed on the four commonly used OCR engines Tesseract, Ocropus, Kraken and Calamari. Figure 5 shows the overall structure of our online training system. Through a web-based front-end, the user can interact with the server-side component. It allows users to upload datasets, OCR models, or a configuration file containing the parameter setting to train, fine tune or evaluate an OCR model.

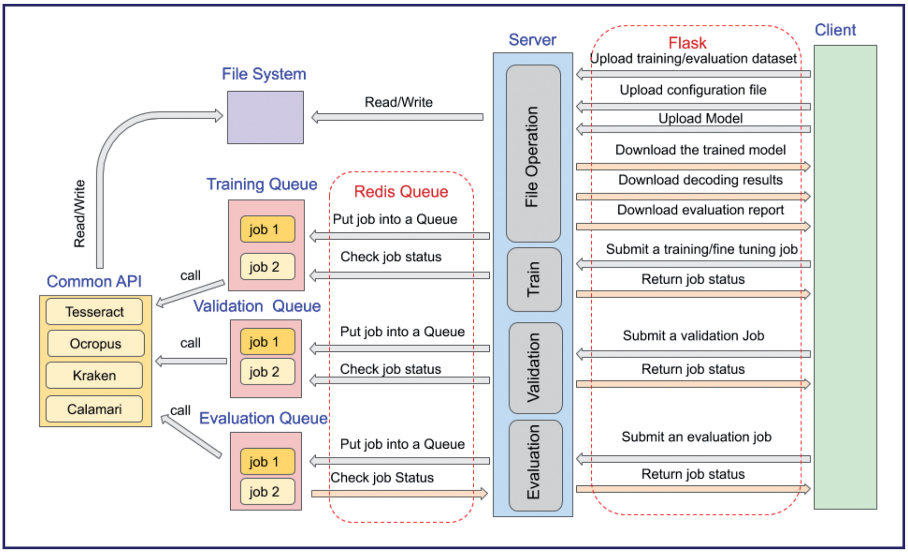


Figure 5. *Framework of the okralact system*

The server-side mainly contains a common API for the above mentioned OCR engines with unified training, decoding and evaluating interfaces (see Fig. 6). For example, a user can now upload one dataset and it is automatically split into a training and a validation set. We also added an evaluation component which compares results with the ground truth files to analyse the performance of the model.

[18] Konstantin Baierer - Rui Dong - Clemens Neudecker, *Okralact - a multi-engine Open Source OCR training system*, in: *Historical Document Imaging and Processing (HIP) 2019, 5th International Workshop 21-22 September 2019, Sydney, Australia*, p. 25–30, <https://doi.org/10.1145/3352631.3352638>.

A very important feature of our system is the unification of the interfaces of the different OCR engines. This allows users to try out the engines without having to learn all of their individual specifics. This harmonisation was a long and complicated task, for several reasons. First, the documentation often does not provide accurate details about parameters, and thus we had to frequently dive into the source code of the engines to see exactly how the parameters are used (and what they correspond to). Second, in many cases the engines have parameters that share similar functions, but have different names and have to be used in a slightly different way (e.g. number of iterations and number of training samples), or in extremely different ways (e.g. defining the network architecture). And, third, some parameters are available only for some of the engines, so our system has to take this into account in order to allow full access to the engines' capabilities.

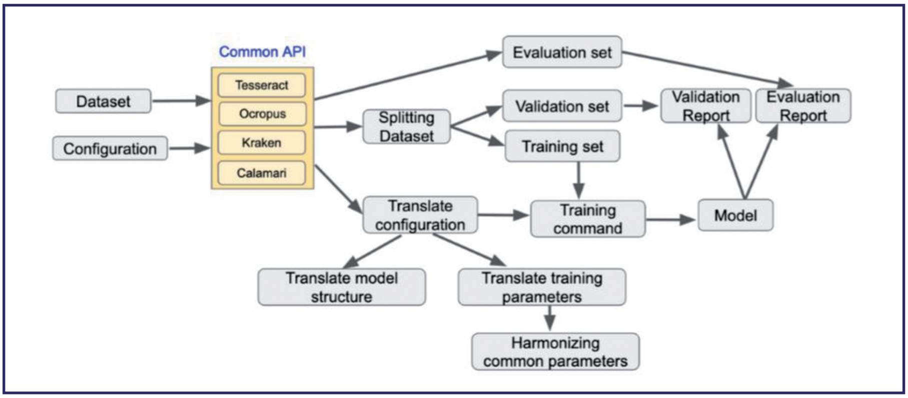


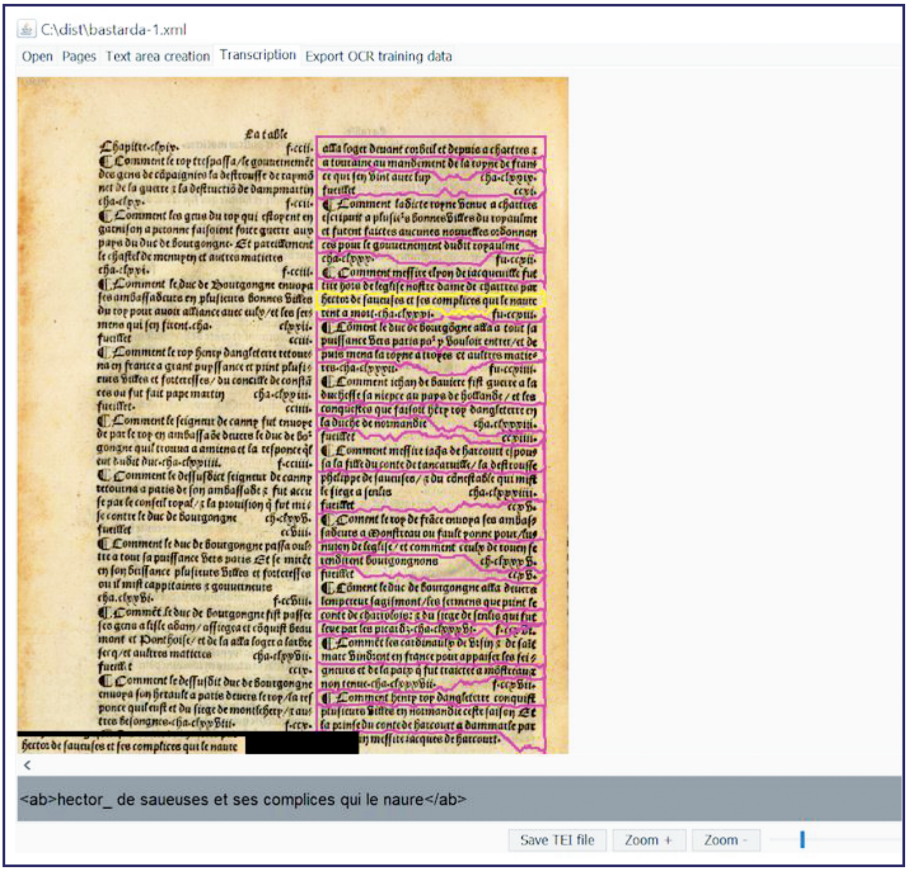
Figure 6. *Workflow of the common API component*

We also created the possibility for users to store and share their models in an online repository. The metadata stored with the models records the training set and parameters. It is also possible to evaluate the model with the help of a test set if the users would like to know how the model performs before using it.

Finally, we transcribed lines for three comparatively difficult font groups for which current OCR models have very high error rates – Schwabacher, Textura and Bastarda. Therefore, we adapted and used DIVAnnotation, a segmentation, labelling, and transcription tool. It is a semi-automatised modular annotation tool originally developed in 2018 by the DIVA group, from the University of Fribourg, Switzerland[19]. Our modifications of this tool mainly consisted in making its inter-

[19] Mathias Seuret - Manuel Bouillon - Fotini Simistira - Marcel Würsch - Marcus Liwicki - Rolf Ingold, *A Semi-automatized Modular Annotation Tool for Ancient Manuscript Annotation*, in: *13th IAPR International Workshop on Document Analysis Systems (DAS)*, Vienna, 2018, p. 340-344, <https://doi.org/10.1109/DAS.2018.80>.

face significantly more intuitive, tune it to our specific needs, and fix several bugs. The adapted version of DIVAnnotation is very easy to use and allows the user to generate transcriptions of historical text very quickly (Fig. 7)[20].



hector de saueuses et ses complices qui le naure

<ab>hector_ de saueuses et ses complices qui le naure</ab>

Figure 7. *DIVAnnotation lite for transcribing historical documents*

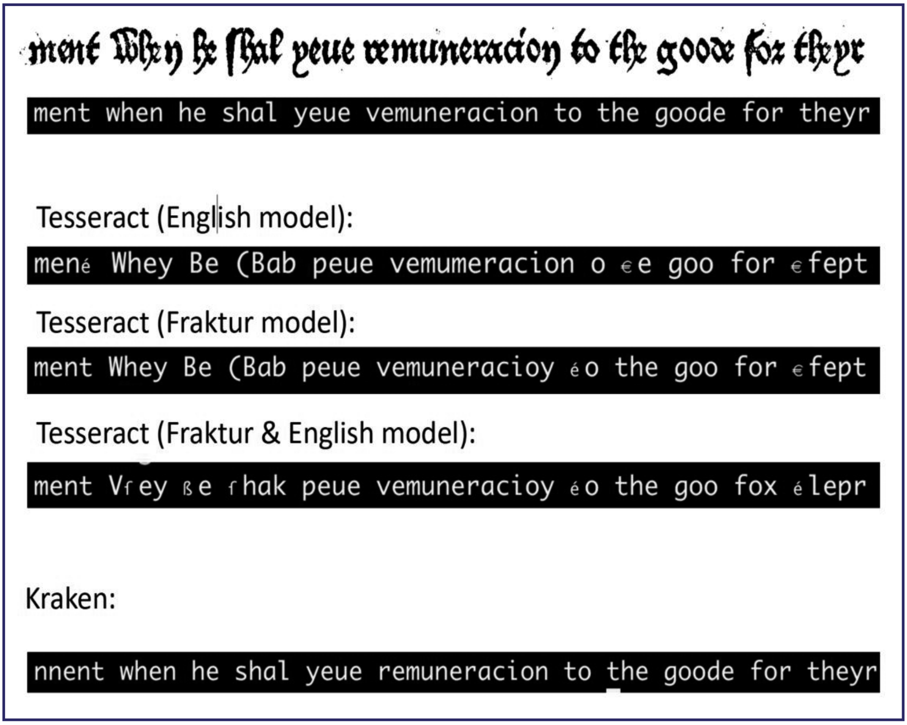
We also designed a test for a rather difficult rare book in the British Library[21]. For his *Dits des philosophes. The dictes or sayengis of the philosophres* (1477), the first English printer, William Caxton, used an exceptional Bastarda[22]. When using Tesseract, we received the following results: for the English model the error rate was 41% (41 out of 100 characters were wrong). The Fraktur model worked slightly better but still produced an error rate of 28%. A combination of the two models produced an error rate of 31%.

[20] https://github.com/seuretm/divannotatio-lite.

[21] We were kindly invited by the British Library to present our results on 18 June 2019.

[22] ISTC id00272000, <https://data.cerl.org/istc/id00272000>.

Next, we used DIVAnnotation to transcribe a few hundred lines and used only 92 lines to train a model for Kraken (the rest was kept for validation). We chose this low number of lines to demonstrate that users can already have good results without putting in too much effort. After all, users presumably do not want to transcribe thousands of lines before they can use OCR on their specific book. When we used our trained model on new text lines of the book, it produced an error rate of only 8%.



ment when he shal yeue vemuneracion to the goode for theyr

Tesseract (English model):

mené Whey Be (Bab peue vemumeracion o €e goo for €fept

Tesseract (Fraktur model):

ment Whey Be (Bab peue vemuneracioy éo the goo for €fept

Tesseract (Fraktur & English model):

ment Vſey ße ſhak peue vemuneracioy éo the goo fox élepr

Kraken:

nnent when he shal yeue remuneracion to the goode for theyr

Figure 8. *OCR results for a difficult Bastarda text, using Tesseract with various models and a pre-trained Kraken model. The textline and its transcription are at the top, the results of the OCR engines with their various models below*

In the past two years, we have published the programs that we developed as open source and provided detailed notes on the code[23]. Thus, our work can easily be used by other researchers and adapted to specific needs. The font group recognition system has already attracted great interest and has been adapted to some needs of the Transkribus team[24]. A trained network is also provided with the recognition system, thus it can be used either as a part of the OCR-D framework, or as a standalone application.

[23] <https://github.com/seuretm/ocrd_typegroups_classifier/>; <https://github.com/Doreenruirui/okralact>.

[24] https://read.transkribus.eu/2019/09/26/printed-vs-handwritten-text-lines-automatically-separated/.

Furthermore, the dataset of our *Dataset of Pages from Early Printed Books* paper is freely available online[25]. It consists of more than 35,000 pages of printed books dating from the 15th to the 18th century, which makes it one of the largest datasets in the field. It is also the very first dataset for font group identification in early printed books. Before, datasets almost exclusively consisted of two major font groups, i.e. Antiqua and Fraktur, with many other Gothic fonts being labelled wrongly as Fraktur as well as no differentiation between Antiqua and Italic. This created major issues for OCR. Hence, we accumulated a much more varied dataset, which represents the diversity of early modern fonts more accurately. Since its publication in August 2019, the dataset has attracted over 400 views and 150 downloads. The pages are randomly selected, unsorted, and therefore cannot be read consecutively to discourage any other use of the pages.

Conclusions

While the large-scale application of font group specific OCR is still hampered by a lack of training data, our project opened the door to this approach, that has remarkable potential. Yet font group recognition also opens new avenues for historical scholarship. We recently started to explore these possibilities by examining the emergence of the font group "Fraktur" in the 16th century and the development that lead to it becoming the most popular font choice for German texts for centuries. To address this issue, we used our tool on some 85,000 German books, printed between 1472-1800, which have been digitised by the Bavarian State Library, the German library with the by far largest collection.

The results largely confirmed previous (but rather anecdotal) observations, showing that Schwabacher, previously the most common font for German language printing, was gradually replaced by Fraktur over the run of the 16th century. In the course of the 17th and 18th century, Fraktur was used in around 9 out of 10 German language books. Contrary to previous assumptions, the 1790s show a slight uptick in the use of Antiqua for German language printing, demonstrating that the Antiqua-Fraktur-dispute had more of an impact than previously assumed[26]. This small study showed how machine learning can help answer questions in the domain of book history that could not have been addressed with conventional methods. We are very optimistic that this tool has many further applications.

[25] https://doi.org/10.5281/zenodo.3366685.

[26] Nikolaus Weichselbaumer - Mathias Seuret - Saskia Limbach - Lena Hinrichsen - Andreas Maier - Vincent Christlein, *The rapid rise of Fraktur*, in: *Digital Humanities im deutschsprachigen Raum (DHd) 2020. 7th International Conference 02-06 March 2020, Universität Paderborn*, p. 229-231, <https://doi.org/10.5281/zenodo.3666690>.

I libri stampati prima del 1800 presentano molte difficoltà per l'OCR. Uno degli ostacoli principali è rappresentato dalla mancanza di diversità dei caratteri storici usati per lo sviluppo del sistema. Il progetto OCR-D, che ha visto la collaborazione di storici del libro e informatici, ha avuto il fine di affrontare questa carenza concentrandosi su tre problemi fondamentali. Il primo obiettivo è stato creare uno strumento che riconoscesse automaticamente le famiglie di caratteri nelle immagini di testi a stampa antichi. Ci siamo concentrati sulle famiglie di caratteri gotici comunemente usate nei testi stampati in Germania nel XV e XVI secolo: il più conosciuto Fraktur e le meno note Bastarda, Rotunda, Textura e Schwabacher. Lo strumento è stato sviluppato su 35.000 immagini, raggiunge un livello di precisione del 98% e riesce a distinguere non solo tra le famiglie di caratteri già menzionate ma anche tra ebraico, greco, antiqua e corsivo. È inoltre in grado di identificare immagini xilografiche e dati irrilevanti (coperte, pagine bianche ecc.). In una seconda fase abbiamo creato un'infrastruttura online (okralact) che facilita l'uso di vari motori OCR open source come Tesseract, OCRopus, Kraken e Calamari e che, allo stesso tempo, facilita l'apprendimento di modelli specifici per famiglie di caratteri. L'elevata precisione di questo software per il riconoscimento apre la strada all'opportunità senza precedenti di distinguere i caratteri utilizzati da ogni stampatore. Con una maggiore quantità di dati per il raffinamento e aggiustamenti successivi, questo strumento può rivelarsi utile nel colmare una lacuna considerevole nella ricerca storica.

L'ultima consultazione dei siti web è avvenuta nel mese di dicembre 2020