Human-Centred Open-Source Automatic Text Recognition for the Humanities with OCR4all

Christian Reul*, Maximilian Nöth, Herbert Baier, Kevin Chadbourne and Florian Langhanki

Centre for Philology and Digitality (ZPD), University of Würzburg, Germany

Abstract

Not least due to the rise of quantitative methods and their need for large volumes of high-quality text, Automatic Text Recognition (ATR) is becoming increasingly important in (Digital) Humanities research. Major technical advances in recent years have been supported by the continuous release of open-source implementations that cover the various individual steps of the ATR workflow. As the optimal approach varies depending on the material at hand and individual quality requirements, the next crucial step is the (further) development of user-friendly software that enables flexible, integrative and sustainable combinations of current and future solutions. To this end, we present the completely revised open-source tool OCR4all, which allows even non-technical users to independently perform an ATR workflow and achieve high-quality results. OCR4all is fully compatible with the ATR solutions from the DFG funding initiative "OCR-D", so that users have a variety of tools at their disposal that they can use in the best possible way for their specific requirements.

Keywords

automatic text recognition, practical OCR/HTR, gui-based tool, open-source

1. Introduction

Recent advancements in Automatic Text Recognition (ATR) have significantly enhanced the capability to extract machine-actionable text (which is searchable, annotatable, suitable for quantitative analysis, etc.) from images, thereby transforming the methodologies employed in humanities research. ATR can be applied in various application scenarios ranging from mass full-text digitization in libraries and archives to the strongly interactive high-quality transcription of textual witnesses. Like various other ongoing developments of AI in general, it offers substantial opportunities to assist humanities scholars in their work, particularly in editorial projects and computer-aided evaluations.

The typical ATR workflow consists of several stages, including key processes such as layout analysis and text recognition, helpful but mostly optional steps like preprocessing and post-correction, and supplementary tasks like consistency checks or format conversions. A common feature of pretty much all of these steps is that there usually is not *the one universally best solution* available as the suitability always depends on the features of each individual use case, such as the material at hand, the quality requirements, the intended subsequent utilization of the results, the available resources etc. In addition, while there are still many capable heuristic approaches around, a clear trend towards trainable methods can be observed. Consequently, users working with ATR are usually always on the hunt for more and better training data. Since the generation of this data generally requires notable manual interaction, sensible approaches to get the *human-in-the-loop* are highly in demand. In this paper we present the fully revised version of the open-source software OCR4all, which addresses the aforementioned needs by combining various freely available ATR processors into a single tool that can be operated via an easy-to-use graphical user interface.

The remainder of this paper is structured as follows: In Sec. 2 we address notable Related Work. Sec. 3 briefly covers the philosophy and history of OCR4all as well as related projects before Sec. 4

Humanities-Centred AI (CHAI), 4th Workshop at the 47th German Conference on Artificial Intelligence, September 23, 2024, Würzburg, Germany

*Corresponding author.

christian.reul@uni-wuerzburg.de (C. Reul); maximilian.noeth@uni-wuerzburg.de (M. Nöth); herbert.baier@uni-wuerzburg.de (H. Baier); kevin.chadbourne@uni-wuerzburg.de (K. Chadbourne); florian.langhanki@uni-wuerzburg.de (F. Langhanki)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-3814/paper7.pdf

introduces its completely reworked version 1.0. Typical use cases and application scenarios of working with OCR4all are covered in Sec. 5 before Sec. 6 concludes the paper.

2. Related Work

In this chapter, we take a brief look at some of the numerous existing ATR solutions. Naturally, the main focus is placed on open-source software, although some (partially) proprietary solutions are included as well.

2.1. Command line-based solution

There is a large number of almost exclusively command line-based tools that cover the most important steps of a complete ATR workflow: These include **OCRopus** [1, 2], **Tesseract** [3], and **Kraken** [4] including its fundamentally re-implemented and fully trainable layout analysis [5].

In addition, there are a number of open-source solutions that focus on a single step of the workflow, including **Eynollah** [6] for layout analysis as well as **Calamari** [7], **PyLaia** [8], or the transformer-based **TrOCR** [9] for text recognition.

2.2. GUI-based solutions and platforms

Several systems offer automatic solutions for layout analysis and text recognition as well as a suitable GUI-based correction interface and the possibility to train new models: Commercial platforms include **Transkribus** [10] and **Teklia**[1], most of whose submodules are proprietary and closed-source, which, among other things, makes it difficult to reuse trained models outside the respective platforms and also raises questions regarding data integrity and sovereignty. **PeroOCR**[2] made some modules and models freely available, but this does not currently appear to include the training functionalities. Finally, to the best of our knowledge, **eScriptorium** [11] is the only fully open-source GUI-based solution besides OCR4all that covers the entire ATR workflow. So far, however, all main automatic steps are covered exclusively by Kraken, which is fundamentally different from our approach.

We conclude that there are a variety of capable and often freely available solutions for individual steps of the ATR workflow, but there is a lack of open-source systems that focus on the combination of these solutions into a powerful and flexible overall package.

3. OCR4all, OCR-D, and OCR4all-libraries

From the beginning, OCR4all's main focus lied on enabling non-technical users to perform ATR on challenging material completely independently and achieve high-quality results. To achieve this, OCR4all, in contrast to most comparable tool, focused on the integration and combination of different freely available ATR software and only supplemented these with additional individual solutions in very specific cases. In addition, OCR4all pursues a very strict and complete open-source policy.

OCR4all has this in common with the coordinated DFG funding initiative for the further development of OCR processes **OCR-D** [12] which aims to prepare the mass full-text transformation of historical prints published between the 16th and 18th century in the German-speaking world in the best possible way. OCR-D's approach is based on interoperability and connectivity, which results in a high degree of flexibility and sustainability. To achieve this numerous solutions for individual workflow steps (>50 ATR *processors*) were integrated into a unified framework, allowing their flexible combination and, consequently, the precise optimization of workflows to individual use case. The project's focus lies on command line usage and all project results are completely open-source.

[1] https://teklia.com
[2] https://pero-ocr.fit.vutbr.cz

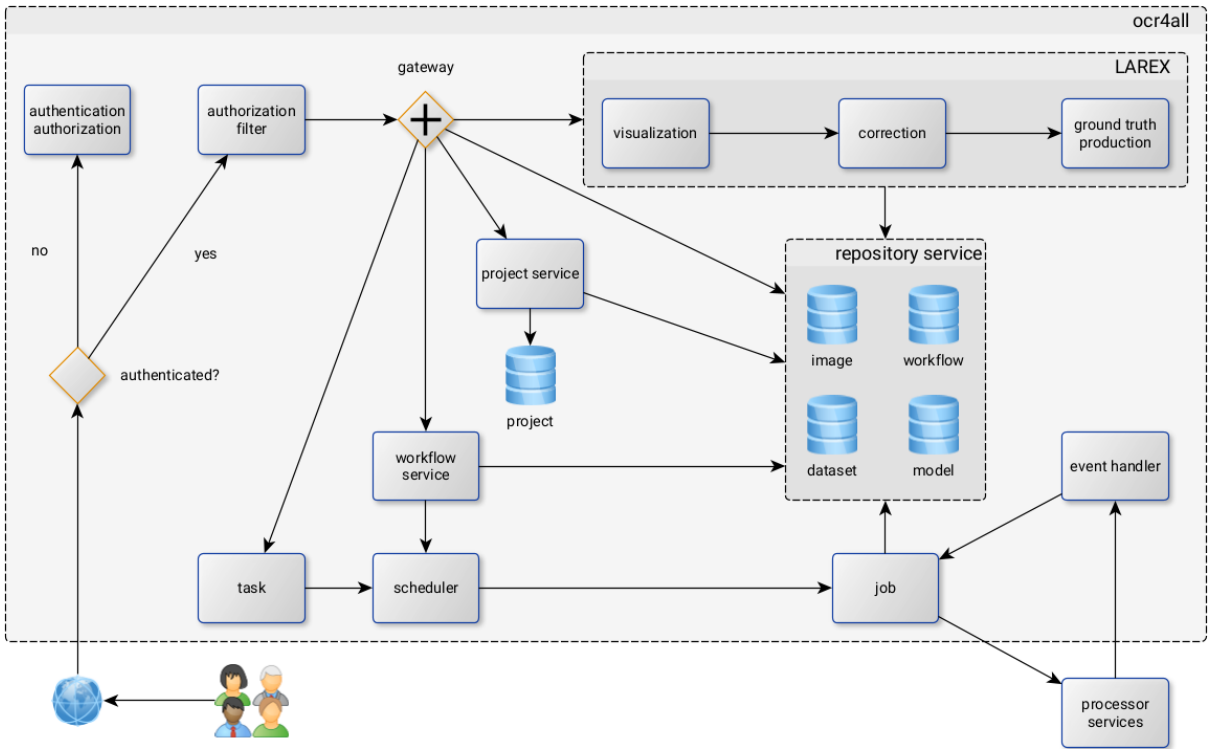

Figure 1: OCR4all system architecture including services and processes.

As OCR4all and OCR-D have many similarities and obviously great synergy potential, the DFG-funded project *OCR4all-libraries* aimed to strengthen the connection of the two approaches. Two main objectives were pursued: 1) The convenient use of OCR-D solutions via OCR4all to ensure independent application by non-technical users. 2) To support the optimization of the ATR result within OCR4all in order to offer added value even to technically experienced users.

4. OCR4all

As the original version of OCR4all became outdated in terms of technology and user requirements as well as the underlying technologies evolved significantly over the years, we used the extensive changes required by the OCR4all-libraries projects as an opportunity to restructure OCR4all from the ground up. The most important features of the resulting revision are presented in the following. After an initial overview of the system's architecture and tech stack, we will introduce the core modules of OCR4all.

4.1. System architecture and technologies used

The entire development is geared towards modularity and interoperability and is based on a strict separation of backend and frontend (see Figure 1). The backend is implemented in Java using the Spring Boot framework, while the frontend is built on the Vue.js ecosystem, with communication facilitated via a REST API.

4.2. Modules

The core objective of the new OCR4all implementation is to provide users with flexible adaptability to numerous heterogeneous application scenarios. This is supported by the following modules:

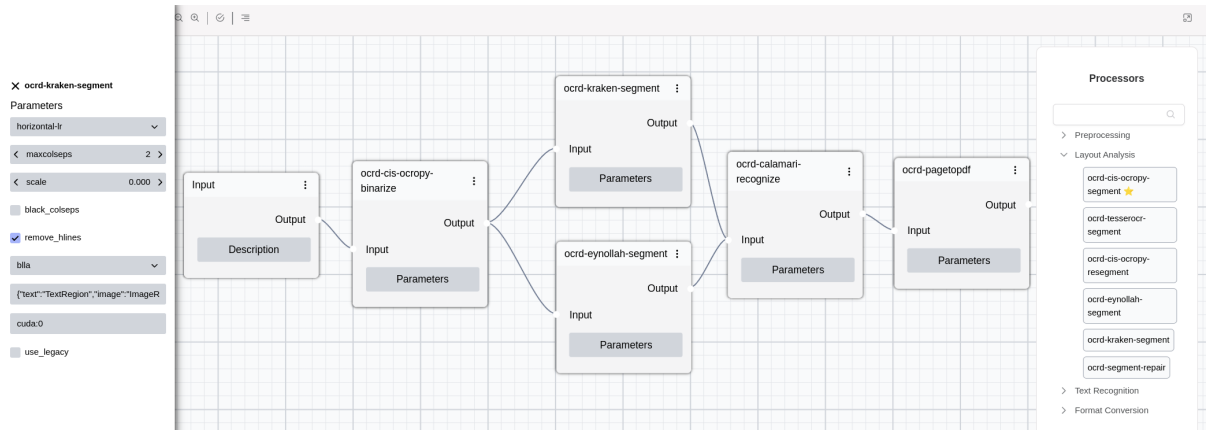

Figure 2: Example workflow in NodeFlow, consisting of different processors, some with extended settings view (left), and the corresponding processor selection (right).

4.2.1. Repository

Each OCR4all instance has a central repository that is used to store and manage all data (images, workflows, models, training/evaluation datasets), thus enabling a strict separation of data management and processing. For individual data, it is possible to add descriptions, assign tags for search and filtering, or generate various statistics, such as the number of available Ground Truth (GT) lines or the codec and its character distribution within a dataset. In addition to entering, changing, and exporting data, it can also be flexibly shared with other users or user groups.

4.2.2. ATR processors and NodeFlow

Numerous ATR processors are available, which can be conveniently and flexibly combined into workflows using the NodeFlow editor (see Figure 2). In addition to linear workflows (one processor for each step), it is also possible to create so-called *branched* workflows to try out different solutions (processors or models) for the same step in order to compare them afterwards to identify the most suitable combination. Additional processors can be integrated either automatically via well-defined OCR-D interfaces or manually.

4.2.3. Result Viewer

After applying a workflow either to all or only selected pages of a project the results are presented in a tree-like view (see Figure 3). In this Result Viewer each processing step of the workflow is represented by a node providing various options for interaction to the users with the goal of maximizing interactivity, explainability, and reproducibility: Firstly, users can execute individual processors from any node, allowing for a more incremental approach, if desired. Additionally, each node contains a transcript of records that is generated for every step of the workflow, containing all parameters and relevant processing metadata in order to maximize reproducibility. Furthermore, the current state of the data (images + PAGE XML) can be exported at any node, added to a dataset (see below), or opened in LAREX for visual review.

4.2.4. LAREX

Naturally, LAREX continues to provide core functionality for visualizing and editing the (intermediate) results of the workflow, including region types, region and line polygons, baselines, reading order, and text recognition results. Due to the flexible combination of processors, LAREX has meanwhile taken on an important role as a visual explanation component that allows a convenient comparison of different results from different workflows for the same page(s).

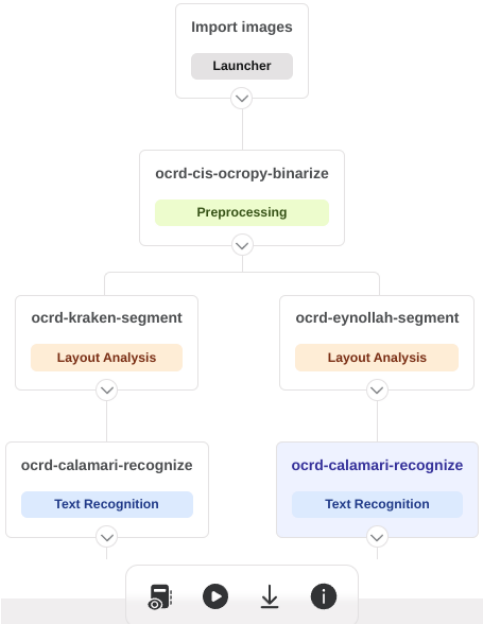

Figure 3: Snippet of the output produced by the example workflow from Figure 2 displayed in the Result Viewer.

4.2.5. Datasets, Training, and Evaluation

Pages from one or several projects can be freely combined into datasets. If required, this can be supported by enriching projects and pages with information, e.g. on age, font, layout type, etc., using the aforementioned tagging functionality, in order to define specific corpora. Datasets and their contained pages can be opened and viewed in LAREX whose extensive correction options can then be put to work to generate GT. This data can then be utilized to evaluate different workflows in a quantitative way as well as to train further models.

5. Practical Usage

In order to meet the requirements of various and their individual needs even better, the aforementioned modules can be used in two independent modes with different focuses which we will introduce in the following. Thereafter we discuss some prototypical example use cases and application scenarios and explain how they can best be addressed using OCR4all.

5.1. One tool, two modes

Due to the requirements of the aforementioned OCR4all-libraries project the initial focus was placed on the so-called **pro** mode which is closely connected to the use cases addressed by OCR-D. It is more exploratory in its application and offers significantly more options and freedom providing unrestricted access to all processors, parameters, and features.

The focus lies on the supported identification of the best workflows/models for specific needs utilizing NodeFlow (configuration), LAREX (qualitative evaluation) and the data set and evaluation modules (quantitative evaluation). Once the most suitable workflow has been determined for the current use case, it can be applied directly to further material, be shared via the repository, or be exported for example to be executed on a different infrastructure.

In contrast, the promptly available **base** mode will offer a highly guided, linear workflow that can also be easily operated independently by non-technical users. In order to reduce complexity, only one (expandable) selection of processors is made available for each step, the parameters are pre-filtered, and there is only restricted access to advanced features.

5.2. Example Use Cases and Application Scenarios

Thanks to the new implementation and the numerous functional enhancements, OCR4all can now be used in a much more flexible way. The best approach is not always clear-cut, as it depends on the material in question, the individual quality requirements, and the available resources. Nevertheless, there are some prototypical use cases that we will briefly discuss in the following.

5.2.1. Fully automatic mass full-text digitalization: Maximal throughput, minimal manual effort

This first use case will most commonly be found in institutions like libraries and archives which aim to process large amounts of scans from their holdings. Since this is basically the prototype of the envisioned OCR-D use case the possibilities of OCR4all in this application scenario have considerably improved due to its full compatibility with OCR-D and the resulting variety of processors. In most cases the most sensible approach will consist of using the pro mode (NodeFlow, LAREX, and datasets) to first identify the most suitable workflow on a few representative test pages of the material at hand or more likely the most suitable workflows for various groups or clusters of similar material.

Then, they can either be executed directly or within the OCR4all instance or, if available, be exported and applied on a dedicated high-performance cluster/server running a standard stand-alone OCR-D installation.

5.2.2. Flawless transcription of source material: Maximum quality, significant manual effort acceptable

In contrast to the first use case, the second one is located at the opposite end of the spectrum regarding automation as well as necessary and acceptable human interaction. For example, when the goal is to prepare a text basis for a digital edition, a perfect transcription result is often expected. Consequently, it is usually essential that the editors carefully check every line and every word manually.

Using the upcoming base mode this process can not only be significantly accelerated but also considerably supported: On the one hand, a so-called *iterative training approach* allows for the continuous use of corrected data as GT to retrain models, constantly enhancing accuracy as more pages are processed. On the other hand, numerous features will support transcription, including a line-synoptic correction view, a freely configurable virtual keyboard for entering special characters, and ongoing consistency checks that provide insight into the characters used.

5.2.3. Building corpora for quantitative applications: Maximizing quality, minimizing manual effort

This third use case is located between the first two and deals with the task of building corpora to train and evaluate quantitative methods or simply apply these methods to. Due to the advance of quantitative methods, the creation of these corpora is becoming increasingly important. Typically, the text quality does not need to be perfect, as most methods are somewhat robust against ATR errors. However, a certain minimum standard is usually expected. Consequently, since large quantities of text are required, the goal is to generate these texts with good but not perfect quality in a very efficient way.

While the aforementioned quality standards leave some room for errors the desired results are often still beyond the current capabilities of available standard models. Correspondingly, building corpora for quantitative analysis using ATR often requires substantial training efforts to produce source-specific models which are geared towards the optimal recognition of one manuscript/hand or one book/printing-type. The goal is to reach the target error rate for the respective source with minimal manual effort, i.e. with minimal GT. The generated model is then applied fully automatically to the remaining pages of the respective source. Moreover, to further increase the effectiveness and efficiency of the entire text production process, it is sensible and almost mandatory to regular retrain the mixed models which represent the starting point when processing each individual source, by combining the newly generated

source-specific training data into one corpus. Unfortunately, this requires a notable amount of data management or rather data housekeeping. Fortunately, this tedious task can be considerably simplified by using OCR4all's datasets and tagging functionality.

6. Conclusion and Future Work

In this paper we introduced OCR4all which is a tool for ATR that is completely open-source, easy to use, and has shown to produce very good results even when applied to demanding historical material regardless of whether they are prints or manuscripts. Due to OCR4all's immense range of functions, it can be used flexibly and allows the best possible adaptation to the diverse requirements and challenges of a broad spectrum of use cases varying considerably regarding quality requirements and achievable degree of automation.

In addition to continuous tasks such as constant refactoring and streamlining, incorporating and addressing user feedback and feature requests, and improving documentation, we plan to add further processors, especially for training and evaluation, mostly for the text recognition step but also for layout analysis. Another important goal is to establish OCR4all as a service for broad institutional use by means of centralized instances. The technical foundations in terms of user administration and resource management are already in place, but there are still some challenges with regard to the connection of external resource management systems and the connection to existing security layers (OAuth/LDAP). Moreover, we aim to enhance interoperability with digitization workflow tools commonly used in stock-holding institutions such as libraries and archives. Concrete steps have already been taken in collaboration with various institutions that use Goobi or Kitodo.

With a view to longer-term and more visionary planning we are eyeing the integration of established as well as cutting edge Natural Language Processing (NLP) processors, such as those made available through HuggingFace, to enhance the in-tool enrichment of the generated machine-actionable texts via typical NLP tasks like Named Entity Recogniton (NER) or Named Entity Linking (NEL). This integration aims to transform raw text into a more structured and semantically meaningful form, significantly expanding the tool's utility for researchers and professionals who require enriched data for downstream tasks like knowledge extraction and content analysis. Furthermore, this will also facilitate the direct integration of the generated results of an OCR4all pipeline into library and archive catalogs, enhancing advanced search capabilities by allowing more precise indexing, metadata generation, and the ability to search for specific entities or concepts within the digitized content. This is of course an ambitious goal but after all, when it comes to research in the area of (Digital) Humanities, the step of ATR does usually not represent the end but merely the beginning.

Acknowledgments

This work has been funded by the German Research Foundation (DFG) under grant number 460665940.

References

[1] T. M. Breuel, The OCRopus open source OCR system, in: Document Recognition and Retrieval XV, volume 6815, International Society for Optics and Photonics, 2008, p. 68150F.

[2] T. M. Breuel, Recent progress on the OCRopus OCR system, in: Proceedings of the International Workshop on Multilingual OCR, ACM, 2009, p. 2.

[3] R. Smith, An overview of the Tesseract OCR engine, in: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), volume 2, IEEE, 2007, pp. 629–633.

[4] B. Kiessling, Kraken - an Universal Text Recognizer for the Humanities, DH 2019 Digital Humanities (2019).

[5] B. Kiessling, A modular region and text line layout analysis system, in: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE, 2020, pp. 313–318.

[6] V. Rezanezhad, K. Baierer, M. Gerber, K. Labusch, C. Neudecker, Document layout analysis with deep learning and heuristics, in: Proceedings of the 7th International Workshop on Historical Document Imaging and Processing, 2023, pp. 73–78.
[7] C. Wick, C. Reul, F. Puppe, Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition, Digital Humanities Quarterly (2018).
[8] J. Puigcerver, C. Mocholí, Pylaia, https://github.com/jpuigcerver/PyLaia, 2018.
[9] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, F. Wei, Trocr: Transformer-based optical character recognition with pre-trained models, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 13094–13102.
[10] P. Kahle, S. Colutto, G. Hackl, G. Mühlberger, Transkribus - a service platform for transcription, recognition and retrieval of historical documents, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 04, 2017, pp. 19–24.
[11] B. Kiessling, R. Tissot, P. Stokes, D. S. B. Ezra, escriptorium: an open source platform for historical document analysis, in: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), volume 2, IEEE, 2019, pp. 19–19.
[12] C. Neudecker, K. Baierer, M. Federbusch, K.-M. Würzner, M. Boenig, V. Hartmann, OCR-D: An end-to-end open-source OCR framework for historical documents, Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage (2019).

A. Online Resources

- OCR4all on GitHub,
- OCR4all Website.