ABSCHLUSSBERICHT

1 Allgemeine Angaben

DFG-Geschäftszeichen: *GE 3075/9-1*

Projektnummer: *460547474*

Titel des Projekts: *Workflow für werkspezifisches Training auf Basis generischer Modelle mit OCR-D sowie Ground-Truth-Aufwertung*

Name(n) des/r Antragstellenden: *Dr. Sabine Gehrlein*

Ltd. Bibliotheksdirektorin der Universitätsbibliothek Mannheim

Dienstanschrift/en: *Universitätsbibliothek, Schloss Schneckenhof, 68161 Mannheim*

Name(n) der Mitverantwortlichen:

Name(n) der Kooperationspartnerinnen und -partner:

Berichtszeitraum (gesamte Förderdauer): *01.07.2021 – 31.12.2023*

2 Zusammenfassung / Summary

Die DFG strebt mit der „Koordinierten Förderinitiative zur Weiterentwicklung von Verfahren der Optical Character Recognition“ (OCR-D) die Transformation der im deutschen Sprachbereich erschienenen Drucke des 16. bis 18. Jahrhunderts (VD 16, VD 17 und VD 18) in maschinenlesbare Form an. Das Projekt war als Modulprojekt Teil der 3. Förderphase und hatte zum Ziel, die Texterkennung durch neue generische Modelle für die eingesetzten OCR-Programme weiter zu verbessern. Zusätzlich sollte es Einrichtungen ermöglichen, die Texterkennung bei Bedarf an einzelne Werke anzupassen, indem sie mit überschaubarem Aufwand die bereitgestellten generischen Modelle werkspezifisch nachtrainieren. Voraussetzung für neue generische Modelle, die besser sein sollten als die bisherigen, waren Trainingsdaten in bestmöglicher Qualität. Mit verbesserten und erweiterten Trainingsdaten konnten neue generische Modelle für die OCR-Programme Kraken, Calamari und Tesseract trainiert werden. Dabei hat das für Kraken trainierte Modell *german_print* inzwischen schon vielfach seine sehr gute Erkennungsqualität und Nachnutzbarkeit bewiesen. Für das gleichnamige Tesseract-Modell ist zu erwarten, dass es seine Vorgängermodelle schon bald ablösen wird. Zusätzlich konnten ein experimentelles Kraken-Modell *german_handwriting* zur Erkennung von Handschriften und ein domainspezifisches Modell *german_newspapers* für Zeitungen trainiert werden. Die neuen Modelle für Kraken und Tesseract eignen sich sehr gut für Nachtrainings mit dem Ziel, die Zeichenerkennung zu erweitern oder die Qualität zu verbessern. Mit der webbasierten Open-Source-Transkriptionsplattform eScriptorium wurden so nach kurzer Einweisung schon zahlreiche Modelle für Kraken auch durch Nutzende ohne spezielle Vorkenntnisse nachtrainiert. Eine noch experimentelle neu entwickelte Erweiterung für eScriptorium bietet die gleiche Funktionalität für Tesseract, so dass in Zukunft auch dafür Nachtrainings einfach möglich werden.

With the DFG funded coordinated "Initiative for Optical Character Recognition Development" (OCR-D), the DFG is aiming to transform VD prints (16th-19th century) into machine-readable form. The project was part of the third funding phase as a module project and aimed to further improve text recognition through new generic models for the OCR programs used. In addition, it should enable institutions to adapt text recognition to individual works if necessary by retraining the generic models provided on a work-specific basis with manageable effort. The prerequisite for new generic models, which had to be better than the previous models, was training data of the best possible quality. New generic models for the OCR programs Kraken, Calamari and Tesseract could be trained with improved and extended training data. The *german_print* model trained for Kraken has already proven its very good recognition quality and reusability many times over. It is expected that the Tesseract model of the same name will soon replace its predecessors. For Kraken, an experimental model *german_handwriting* could also be trained to recognize handwritten text. A domain-specific model *german_newspapers* was trained for newspapers. The new models for Kraken and Tesseract are very well suited for follow-up training with the aim of improving character recognition or quality. With the web based open-source transcription platform eScriptorium, numerous models for Kraken have

already been retrained, even by users without any special prior knowledge after a brief introduction. A newly developed extension for eScriptorium, which is still experimental, offers the same functionality for Tesseract, so that retraining will also be possible for Tesseract in the future.

3 Arbeits- und Ergebnisbericht

3.1 Ausgangslage und Zielsetzung des Projekts

Ein Hauptziel des Projektes war es, Bibliotheken mithilfe eines nutzerfreundlichen, werkspezifischen Trainingsworkflows in die Lage zu versetzen, die Texterkennung für eigene Bestände zu optimieren. Neben dem Nachtrainings-Workflow standen die Erstellung und Aufwertung hochwertiger Ground Truth (GT) im Fokus des Projekts. Darüber hinaus sollten softwaretechnische Werkzeuge und Anleitungen bei der Produktion, Korrektur und Aufwertung von Ground-Truth-Korpora unterstützen. Ein zentrales, öffentliches Modellrepositorium sollte außerdem die leichte Auffindbarkeit und Weiternutzung trainierter Modelle sicherstellen.

3.2 Arbeitsschritte im Berichtszeitraum

Die Projektziele wurden vollständig erreicht. Die im Antrag vorgesehenen Arbeitsschritte konnten weitgehend wie geplant durchgeführt werden. Einige Anpassungen waren aufgrund neuer Erkenntnisse oder geänderter Rahmenbedingungen erforderlich. Die gewählte Vorgehensweise und die erzielten Ergebnisse werden im Folgenden detailliert beschrieben.

3.2.1 AP 1: Auswahl und Qualifizierung geeigneter Ground-Truth-Daten

Im Projektverlauf wurden existierende Ground-Truth-Korpora weiterentwickelt und neue erstellt. Die hierfür verwendeten Transkriptionsrichtlinien basieren auf den während der Projektlaufzeit erarbeiteten OCR-D GT Richtlinien im Level 2[1], die in enger Abstimmung mit dem OCR-D-Koordinierungsteam bei den 14-tägigen GT-Calls erarbeitet worden sind. Dies ist relevant, da die Konsistenz der Trainingsdaten auch die Qualität trainierter Modelle beeinflusst. Inkonsistente Trainingsdaten führen zu inkonsistenten Ergebnissen oder können die Konvergenz im Trainingsprozess verhindern.

Die schon vorhandenen Ground-Truth-Datensätze *GT4HistOCR*[2], *Fibeln*[3] (18.–19. Jhd.) und *Weisthuemer*[4] (19. Jhd.) konnten erweitert und verbessert werden. Der Datensatz *digitue-gt*[5] (16.–20. Jhd.) entstand 2022 zum Abschluss des Landesprojekts OCR-BW und wurde anschließend weiter verbessert. Neu angelegt wurden die Datensätze *digi-gt*[6] (16.–17. Jhd.) und *dach-gt*[7] (15.–20. Jhd.).

[1] https://ocr-d.de/de/gt-guidelines/trans/
[2] https://code.bib.uni-mannheim.de/ocr-d/GT4HistOCR. Originaldatensatz: Springmann, U., Reul, C., Dipper, S., & Baiter, J. (2018). GT4HistOCR: Ground Truth for training OCR engines on historical documents in German Fraktur and Early Modern Latin (1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.1344132
[3] https://github.com/UB-Mannheim/Fibeln
[4] https://github.com/UB-Mannheim/Weisthuemer
[5] https://github.com/UB-Mannheim/digitue-gt
[6] https://github.com/UB-Mannheim/digi-gt
[7] https://github.com/UB-Mannheim/dach-gt

Die Verbesserungen der genannten GT-Korpora umfassten gängige Verwechslung wie beispielsweise von „rundem s“ [s] und „langem s“ [ſ] sowie die Korrekturen weiterer Transkriptions- und Layoutfehler und sind in der Versionsgeschichte und der Dokumentation der Datensätze beschrieben. Die Überarbeitung der GT-Korpora wurde in Transkribus[8] und eScriptorium[9] (siehe AP 5) realisiert.

GT-Sammlung	Zeitraum	Seiten	Textzeilen	Unicode-Zeichen
Fibeln	1782–1898	453	8.895	306.521
digi-gt	1507–1602	115	3.240	145.155
digitue-gt	1521–1877	273	9.403	574.459
Weisthuemer	1869	35	1.973	91.084
dach-gt	1486–1913	117	3.412	133.983
GT4HistOCR	1471-1898	-	313.204	13.893.510
Gesamt		993	340.127	15.144.712

Tabelle 1: Ground-Truth-Korpora (Bücher und Zeitschriften)

Weitere im Projekt erstellte bzw. aufgewertete Ground Truth basiert auf historischen Zeitungen und Amtsschriften. Der Fokus auf Zeitungen wurde gewählt, um auf aktuelle Bedarfe der Forschung zu reagieren. So zeigte sich zum Projektstart ein erhöhtes Interesse an Zeitungs-Ground-Truth und entsprechender Modelle. Ziel war es deshalb, neben hochwertiger Zeitungs-GT ebenso hochwertige generische Modelle zu trainieren. Die auf Basis der erstellten bzw. aufgewerteten Ground Truth trainierten Modelle werden im AP 3 beschrieben.

Die in einem iterativen Prozess neu erstellten GT-Korpora umfassen die r*eichsanzeiger-gt*[10], die *hkb-gt*[11] sowie das *Charlottenburger Amtsschrifttum*[12]. Auf eine Layoutkorrektur (Regionen, Zeilenmasken und Baselines) folgte die Erstkorrektur der Transkription, die auf einer Pre-OCR basierte. Diese Erstkorrektur wurde anschließend mit einem nachtrainierten Modell verglichen und eine zweite Korrekturschleife durchgeführt. Anschließend folgte eine Qualitätssicherung.

GT-Sammlung	Zeitraum	Seiten	Textzeilen	Unicode-Zeichen
reichsanzeiger-gt	1820–1939	197	119.429	2.967.316
hkb-gt	1931–1945	38	13.420	531.772
Charlottenburger Amtsschrifttum	1891–1910	27	3.576	133.502
Gesamt		262	136.425	3.632.590

Tabelle 2: Neu erstellte Ground-Truth-Korpora für historische Zeitungen und Amtsschriften

[8] https://readcoop.eu/de/transkribus/
[9] https://gitlab.com/scripta/escriptorium
[10] https://github.com/UB-Mannheim/reichsanzeiger-gt
[11] https://github.com/UB-Mannheim/hkb-gt
[12] https://github.com/UB-Mannheim/charlottenburger-amtsschrifttum

Die bestehenden Korpora umfassen die *AustrianNewspapers*[13] und die *NZZ Black Letter Ground Truth*[14], zwei Korpora, an denen die Universitätsbibliothek Mannheim bereits in der Vergangenheit Aufwertungen vorgenommen hat. Neben einer Layoutkorrektur wurde ein Struktur-Labeling[15] für alle Seiten durchgeführt. Die bestehenden Transkriptionen wurden danach in das OCR-D GT Richtlinien Level 2 aufgewertet.[16]

GT-Sammlung	Zeitraum	Seiten	Textzeilen	Unicode-Zeichen
Austrian Newspapers	1864–1911	161	59.851	2.226.879
NZZ Black Letter GT	1780–1946	167	43.328	2.030.336
Gesamt		328	103.279	4.257.215

Tabelle 3: Aufgewertete Ground-Truth-Korpora für historische Zeitungen und Amtsschriften

Die Bearbeitung der Korpora wurde von wissenschaftlichen Hilfskräften von 01/2022 bis 06/2023 in Transkribus durchgeführt, da das Tool zum Projektstart als einziges die Erfassung von Tabellenstrukturen ermöglichte, die einen wesentlichen Bestandteil historischer Zeitungen bilden. Durch die Erfassung dieser Strukturen konnte ein wichtiger Beitrag zur aktuellen Forschung geleistet werden.

Mit Ausnahme des zeilenbasierten Datensatzes *GT4HistOCR* stellen alle Datensätze die GT im Format PAGE-XML bereit.

3.2.2 AP 2: Qualifizierung von softwaretechnischen Werkzeugen zur Aufwertung und Korrektur von GT-Datensätzen

Anders als im Projektantrag festgehalten, wurde die Weiterentwicklung der von der Universitätsbibliothek Mannheim entwickelten Tools *GTReval*[17] und *GTCheck*[18] zugunsten des neu entwickelten Tools *PagePlus*[19] in Verbindung mit der Transkriptionsplattform eScriptorium aufgegeben. Gründe für diese Änderung lagen einerseits in der Konzeption des Nachtrainings-Workflows, der auf eScriptorium basiert (siehe AP 5), und andererseits in der Notwendigkeit, neben zeilenbasierter auch PAGE–XML-Ground-Truth verarbeiten zu können. Die Features, die *GTCheck* zur Verfügung stellte (zeilenbasierte Korrektur und Aufwertung von GT), sind bereits standardmäßig in eScriptorium

[13] https://github.com/UB-Mannheim/AustrianNewspapers. Originaldatensatz: Mühlberger, G. & Hackl, G. (2019). NewsEye / READ OCR training dataset from Austrian Newspapers (19th C.) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.3387369

[14] https://github.com/UB-Mannheim/NZZ-black-letter-ground-truth. Originaldatensatz: Ströbel, P., Clematide, S. & Weil, S. (2022). impresso/NZZ-black-letter-ground-truth: Improved Test Set. [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7053264

[15] Verwendete Strukturtypen: *header, heading, paragraph, reference, footer*

[16] Übersichten zu den Aufwertungen der *AustrianNewspapers* und *NZZ Black Letter GT* finden sich auf GitHub: https://github.com/UB-Mannheim/AustrianNewspapers/wiki/ReleaseNotes#200 und https://github.com/UB-Mannheim/NZZ-black-letter-ground-truth/wiki/ReleaseNotes

[17] https://github.com/JKamlah/GTReval

[18] https://github.com/UB-Mannheim/GTCheck

[19] https://github.com/UB-Mannheim/pageplus

integriert. Auch die Funktionen von *GTReval* (automatische Evaluation und Aufwertung von GT) sind in eScriptorium in Verbindung mit *PagePlus* enthalten.

PagePlus ist ein Python-basiertes Kommandozeilen-Tool für die Verarbeitung und Analyse von PAGE XML-Dateien. Mit seinen vielfältigen Funktionen ermöglicht es die effiziente Verarbeitung von text- und dokumentregionbasierten Informationen. Das Tool kann Statistiken erstellen, Validierungen durchführen, häufige Fehler automatisch korrigieren sowie Layoutinformationen und Textinhalte modifizieren und bietet verschiedene Exportformate an. Die Konzeption des Tools erlaubt zudem eine einfache Erweiterung um projektspezifische Funktionalitäten. *PagePlus* ist somit nicht nur ein Werkzeug zur Optimierung von Trainingsdaten, sondern kann auch dazu genutzt werden, um wesentlich zur Verbesserung der Texterkennung beizutragen.

Für die Extraktion von Textzeilenpaaren, wie sie unter anderem für das Training von Tesseract-Modellen benötigt werden, wurde das OCR-D Skript *page2img*[20] erweitert und optimiert. Das Skript kann nun auch die Bilder von Textzeilen bei Bedarf um 90, 180 oder 270 Grad drehen und so Trainingsfehler vermeiden.

Um die Vor- und Nachteile der Anwendung von synthetischen und realen Trainingsdaten zu überprüfen[21], wurde die *text2image* Funktion von Tesseract erweitert[22] und das Skript *degrade images* für die zufällige Veränderung der Bilder[23] geschrieben.

3.2.3 AP 3: Training von generischen Modellen und Evaluierung geeigneter Netzwerktopologien

Im AP3 wurden ein generisches Basismodell *german_print*, das das gesamte VD-Spektrum abdeckt, und ein generisches Zeitungsmodell *german_newspapers* für Kraken und Tesseract trainiert. Außerdem wurden verschiedene Netzwerktopologien evaluiert. Da die Arbeiten an den Zeitungstrainingsdaten frühzeitig abgeschlossen werden konnten, wurde die Evaluierung der Netzwerktopologien anhand von diesem generischen Zeitungsmodell durchgeführt. Darüber hinaus wurden zwei Kraken-Segmentierungsmodelle trainiert, die für die Layoutanalyse von modernen und historischen einspaltigem (*ubma_segmentation*)[24] und zweispaltigem (*historical_reports_2col*)[25] Fließtext eingesetzt werden können.

Die Evaluierung erfolgte mit Kraken, das umfassende Möglichkeiten zur Realisierung unterschiedlichster Netzwerktopologien bietet und bei der die Trainingsprozesse selbst deutlich performanter

[20] https://github.com/OCR-D/format-converters/blob/master/page2img.py
[21] https://github.com/UB-Mannheim/charlottenburger-amtsschrifttum/wiki/Work-specific-training-with-Charlottenburger-Amtsschrifttum#about
[22] https://github.com/JKamlah/tesseract/tree/text2imageExtension
[23] https://github.com/UB-Mannheim/charlottenburger-amtsschrifttum/blob/main/scripts/degrade_images/degrade_images.py
[24] https://github.com/JKamlah/ubma-segmentation-ocr-model
[25] https://github.com/JKamlah/historical-reports-2col-ocr-model

ablaufen als beispielsweise bei Tesseract. Da die meisten Topologien keine festen Namen besitzen, sind die verwendeten Namen projektspezifisch und dienen lediglich der besseren Orientierung. Das Training und die Auswertung der Topologien sind ausführlich auf GitHub protokolliert.[26]

Zunächst wurden die in der Kraken-Community am häufigsten verwendeten und empfohlenen Netzwerktopologien identifiziert: die von Benjamin Kiessling vorgeschlagene Topologie *kraken* und die Topologie *htru* aus dem Projekt HTR-United[27]. Außerdem wurde die Topologie *htr+* ausgewählt, die im Transkribus-Projekt[28] sehr erfolgreich eingesetzt wurde. Im nächsten Schritt wurden diese Topologien zusammen mit der VGSL-Spezifikation[29] in das Programm ChatGPT[30] eingelesen und daraus eine völlig neue Topologie *gpt* abgeleitet. Im letzten Schritt wurde eine weitere Topologie *sgd* auf Basis der bestehenden Topologien und aktueller Forschungsprojekte* (DAN) entwickelt. Zum weiteren Vergleich wurde mit diesen Daten auch ein Tesseract-Modell *tesseract* trainiert. Für das Training dieser Topologien wurden die neu erstellten Zeitungstrainingsdaten verwendet: *reichsanzeiger-gt*, *AustrianNewspapers* und *NZZ Black Letter Ground Truth.* Der Trainingsdatensatz *hkb-gt* diente als Evaluationsdatensatz.

	kraken	**htru**	**htr+**	**gpt**	**sgd**
Gesamtgröße (in MB)	16	22	19	31	24
Zeichengenauigkeit (in %)	99,40	99,40	99,41	99,40	99,40
Wortgenauigkeit (in %)	97,20	97,20	97,30	97,00	97,10
Trainingsdauer (in Std.)	51	42	39	60	48

Tabelle 4: Trainingsdaten und -ergebnisse verschiedener Netzwerktopologien

	kraken	**htru**	**htr+**	**gpt**	**sgd**	**digitue (Kraken)**	**tesseract**	**frak2021 (Tesseract)**
Zeichengenauigkeit (in %)	99,61	99,60	99,55	99,53	99,54	98,69	98,75	98,15

Tabelle 5: Evaluationsergebnisse der verschiedenen Netzwerktopologien

Die ausgewählten Topologien schneiden bei der Aufgabe alle sehr gut ab, wobei die *kraken* Topologie am besten abschneidet. Ob sich die Zahlen bei komplexeren Trainingsdaten verschieben, sollte bei zukünftigen Trainings überprüft werden. Für Print-Trainingsdaten, die nur eine überschaubare Variation an Schriften und Glyphen enthalten, ist die *kraken* Topologie aufgrund ihrer Größe und der erreichten Genauigkeiten zu empfehlen. Im Vergleich zu den bisher an der Universitätsbibliothek Mannheim am häufigsten eingesetzten Modellen *digitue* für Kraken und *frak2021* für Tesseract zeigt das Training mit den neuen Zeitungstrainingsdaten eine deutliche Reduktion der Fehler von über

[26] https://github.com/UB-Mannheim/kraken/wiki/Training-German-Newspapers
[27] https://github.com/HTR-United
[28] https://readcoop.eu/de/transkribus/
[29] https://kraken.re/3.0/vgsl.html
[30] https://chat.openai.com/

50 % auf noch unbekannten Zeitungsseiten.

Die beiden generischen Modelle wurden mit der Kraken-Topologie trainiert und auf Zenodo, dem für Kraken-Modelle empfohlenen Repositorium, hochgeladen. Ebenso wurden diese Kraken-Modelle und die entsprechenden Tesseract-Modelle in einem GitHub-Repositorium gemäß AP 4 veröffentlicht. Die Segmentierungsmodelle sind ebenfalls auf Zenodo und GitHub verfügbar.

3.2.4 AP 4: Bereitstellung eines zentralen Repositoriums für Modelle

In Abstimmung mit dem OCR-D-Koordinationsprojekt wurden GitHub, Zenodo und HuggingFace[31] als Ort für das zentrale Modell-Repositorium in Betracht gezogen. Die Entscheidung fiel für GitHub, da Grundvoraussetzungen wie Versionierung, Kostenfreiheit und große Community gegeben waren und da die GT-Datensätze im OCR- und HTR-Umfeld meistens ebenfalls auf GitHub publiziert werden. Die im Kontext OCR-D entwickelten Templates für GT-Datensätze konnten für Modelle entsprechend angepasst werden: ein OCR-Modell Repo Template[32] analog zum GT Repo Template[33] und OCR-Modell Metadata[34] analog zu GT-Metadata[35].

Wenn Modelle auf GitHub oder Zenodo hochgeladen werden, fehlen häufig relevante beschreibende Informationen *Metadaten*. Dieses Problem stellt sich auf der Plattform HuggingFace in der Regel nicht, da sich hier die sogenannten Model-Cards[36] bereits als De-facto-Standard etabliert haben. Allerdings sind die Tools zur Erstellung von Model-Cards sehr unspezifisch und bieten dem Anwender nur eine eingeschränkte Unterstützung. Aus diesem Grund wurde zunächst das OCR Modell Metadata Repositorium mit einer benutzerfreundlichen, standardisierten Online-Eingabemaske[37] zur Erfassung der OCR-Modellmetadaten entwickelt. Die neue OCR-D-Eingabemaske führt den Benutzer wesentlich zielgerichteter durch den Prozess und bietet bereits eine Vielzahl von Vorschlägen und zusätzlichen Hilfestellungen bei offenen Fragen. Während des Eingabeprozesses werden Informationen wie die verwendete GT, die verwendete OCR-Engine und weitere Details zum Training abgefragt. Die gesammelten Informationen können dann per Export als YAML-Datei gespeichert und dem Modell-Repositorium hinzugefügt werden. Dabei spielt die Plattform zunächst keine Rolle, die Metadatendatei kann also jeder Modellpublikation beigefügt werden. Das Schema der Metadatendatei orientiert sich an den Model-Cards von HuggingFace und ist mit diesen weitgehend konform. Um auch das Publizieren der Modelle zu erleichtern und zu vereinheitlichen, wurde zusätzlich das OCR Model Repo Template erstellt. Das Repositorium selbst kann als Template bei der Erstellung eines neuen Modell Repositoriums verwenden werden und legt damit nicht nur die Ordnerstruktur fest, sondern kann in Kombination mit den Metadaten eine automatische Übersichts-

[31] https://huggingface.co/
[32] https://github.com/UB-Mannheim/ocr-model-repo-template
[33] https://github.com/OCR-D/gt-repo-template
[34] https://github.com/UB-Mannheim/ocr-model-metadata
[35] https://github.com/tboenig/gt-metadata
[36] https://huggingface.co/docs/hub/model-cards
[37] https://jkamlah.github.io/ocr-model-metadata/document-your-ocr-model.html

Homepage der Modelle generieren. In einem letzten Schritt können diese OCR-Modell-Repositorien an einem zentralen Ort zusammengeführt und zugänglich gemacht werden. Für die Modelle der Universitätsbibliothek Mannheim dient dazu das Repositorium OCR-Modell-Catalogue.[38]

Da die Zenodo OCR/HTR Model Repository Community[39] bereits offiziell für Kraken-Modelle vorgesehen ist, wird deren Veröffentlichung dort empfohlen und von der Universitätsbibliothek Mannheim zusätzlich genutzt.

3.2.5 AP 5: Implementierung des Nachtraining-Workflows

Die Implementierung des Nachtrainingsworkflows sollte für die OCR-D-relevanten OCR-Engines Tesseract, Calamari und Kraken realisiert werden. Dabei stand ein benutzerfreundlicher Workflow im Vordergrund, der in drei Szenarien, nämlich sowohl lokal als auch über einen Webdienst oder bei einem Dienstleister funktionieren muss.

Ursprünglich war geplant, die bereits vorhandene prototypische Implementierung eines Training-Workflows *Okralact* praxistauglich zu machen und um eine anwenderfreundliche Web-Schnittstelle zu erweitern. Allerdings sah das Projektteam nach Rücksprache mit dem OCR-D-Koordinierungsteam schon früh keinen Vorteil im Ansatz von *Okralact*, das Training der unterschiedlichen OCR-Software auf eine gemeinsame API abzubilden. Zusätzlich hatten die Gutachten zum Antrag dem Projekt empfohlen, kein eigenes Web-User-Interface zu entwickeln.

Deshalb untersuchte das Projektteam, ob ein anwenderfreundlicher Trainingsworkflow auf Basis der schon existierenden Transkriptionsplattformen OCR4All oder eScriptorium realisierbar wäre.

Sowohl OCR4All als auch eScriptorium nutzen für OCR-D relevante OCR-Engines zur Texterkennung, sind Open Source, kostenlos und verfügen über aktive Nutzer- und Entwicklergemeinschaften. Darüber hinaus können beide Tools sowohl auf lokalen Rechnern als auch auf einem Server betrieben werden und verfügen über eine grafische Benutzeroberfläche, was die Benutzerfreundlichkeit deutlich erhöht.

Die Entscheidung fiel auf eScriptorium, da OCR4All sich zu Beginn und während des Projekts in einer weitreichenden technologischen Überarbeitungsphase befand. Tesseract und Calamari konnten als Prozessoren in den Erkennungs- und Trainingsprozess von eScriptorium eingebaut werden. Allerdings verwendet OCR-D eine veraltete, nicht weiterentwickelte Version von Calamari, was so gravierende Auswirkungen hatte, dass die weitere Integration dieser OCR-Engine in eScriptorium gestoppt und der Schwerpunkt auf die Integration von Tesseract verlagert wurde. Der bereits entwickelte Code für die Calamari-Integration ist aber veröffentlicht und steht somit als Basis für eine Weiterentwicklung zur Verfügung, wenn OCR-D auf eine neuere Version von Calamari wechseln

[38] https://github.com/JKamlah/ocr-model-catalogue
[39] https://zenodo.org/communities/ocr_models/

sollte.

Mit der erfolgreich implementierten Erweiterung des Trainingsworkflows haben Anwender*innen in Zukunft die Möglichkeit, werkspezifische Trainings nicht nur für Kraken, sondern auch für Tesseract (und für Calamari mit den erwähnten Einschränkungen) durchzuführen und bereits existierende Texterkennungsmodelle dieser OCR-Engines in eScriptorium zu nutzen. Ein Trainingsleitfaden (siehe AP 6) beschreibt den Nachtrainingsworkflow ausführlich.

Alle drei angestrebten Workflow-Szenarien konnten durch die Wahl von eScriptorium realisiert werden. eScriptorium kann lokal installiert oder auf einem eigenen Server als Web-Applikation genutzt werden, und die Universitätsbibliothek Mannheim hat bereits während der Projektlaufzeit ihre eigene öffentliche nutzbare eScriptorium-Instanz genutzt, um für kleinere Projektvorhaben als Dienstleister Dokumente zu erkennen und neue, werkspezifische Modelle zu trainieren.

Alle notwendigen Codeänderungen sind in Entwicklungszweigen auf GitHub publiziert. Einrichtung und Anwendung für eine erweiterte Instanz von eScriptorium werden bis zur Integration dieser Codeänderungen in die betroffenen Open-Source-Projekte in einer Anleitung erklärt.[40]

3.2.6 AP 6: Erstellung von Anleitungen, Empfehlungen und der Dokumentation

Sowohl die im Projektverlauf entwickelten bzw. erweiterten softwaretechnischen Werkzeuge (PagePlus, eScriptorium) als auch die erstellten GT-Korpora sind auf GitHub dokumentiert (siehe AP 1 und AP 2). Für den werkspezifischen Nachtrainingsworkflow ist ein umfangreicher Trainingsleitfaden auf GitHub bereitgestellt.[41] Dieser umfasst auch Best-Practice-Empfehlungen zur Durchführung eines werkspezifischen Trainings und zur Erstellung eigener Ground-Truth, in der die OCR-D GT Guidelines eingeflossen sind, an deren Erarbeitung das Projektteam aktiv mitgewirkt hat. Weitere trainingsspezifische Anleitungen und Dokumentation für die projektrelevanten OCR-Engines Tesseract, Kraken und Calamari sind ebenfalls auf GitHub veröffentlicht.[42]

3.2.7 AP 7: Projektmanagement und Kommunikation

Projektmitarbeiter nahmen an allen OCR-Entwicklerworkshops und den Treffen der Modulprojekte teil und haben dort über den aktuellen Projektstand berichtet. Außerdem konnten Projektergebnisse auf dem 110. und 111. Deutschen Bibliothekartag 2022/23 sowie auf der Jahrestagung der European Library Automation Group (ELAG) 2022 in Riga vorgestellt werden. Dabei waren die angebotenen Workshops zu eScriptorium (Einführung und Training) ausgebucht, es gab viele sehr positive Rückmeldungen, und etliche der Teilnehmenden nutzten den Testzugang zur Plattform noch Wochen danach oder beantragten einen persönlichen Zugang.

---

[40] https://github.com/UB-Mannheim/eScriptorium_Dokumentation/blob/main/eScriptorium-with-tesseract-extension.md
[41] https://github.com/UB-Mannheim/eScriptorium_Dokumentation/blob/main/Training-with-eScriptorium.md
[42] https://github.com/UB-Mannheim/Projects/tree/main/OCR-D

4 Öffentlich zugängliche Projektergebnisse

4.1 Publikationen mit wissenschaftlicher Qualitätssicherung

- Schmidt, T., & Kamlah,J., & Weil, S. (2024). Reichsanzeiger-GT: An OCR ground truth dataset based on the historical newspaper “Deutscher Reichsanzeiger und Preußischer Staatsanzeiger” (German Imperial Gazette and Prussian Official Gazette) (1819–1945). Data in Brief, Volume 54, 2024, ISSN 2352-3409. https://doi.org/10.1016/j.dib.2024.110274

4.2 Weitere Publikationen und öffentlich gemachte Ergebnisse

Eine Auflistung aller veröffentlichten Projektergebnisse (Software, Ground-Truth-Datensätze, (Trainings-)Dokumentationen und trainierte OCR-Modelle) ist auf GitHub zu finden.[43]

Präsentationen

- Kamlah, J., & Schmidt, T. (2023). Hands-on Lab Ground Truth-Erstellung und Modelltraining mit eScriptorium, 111. BiblioCon Hannover 2023
- Kamlah, J., & Schmidt, T. (2022). Der Weg zum nutzbaren Volltext. Werkspezifisches Training als Baustein der OCR-Volltexterkennung für Alte Drucke. 8. Bibliothekskongress Leipzig 2022: #FreiräumeSchaffen, BID, Leipzig, Germany. https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/17861
- Kamlah, J. & Schmidt, T. (2022). Finetune your OCR! Improving automated text recognition for early printed works by finetuning existing Tesseract models. ELAG 2022, Riga, Latvia. https://dom.lndb.lv/data/obj/file/33561874.pdf

Arbeitspapiere

- Kamlah, J., & Schmidt, T. (2023). Transkriptionsregeln und Guidelines zur Layoutbearbeitung im DFG-Projekt "Workflow für werkspezifisches Training auf Basis generischer Modelle mit OCR-D sowie Ground-Truth-Aufwertung". Zenodo. https://doi.org/10.5281/zenodo.10203335

5 Weitere Informationen zum Projekt

Dieser Teil des Berichtes ist nicht öffentlich.

6 Anlagen mit Programmspezifischen Informationen

Nicht zutreffend.

[43] https://github.com/UB-Mannheim/Projects/tree/main/OCR-D