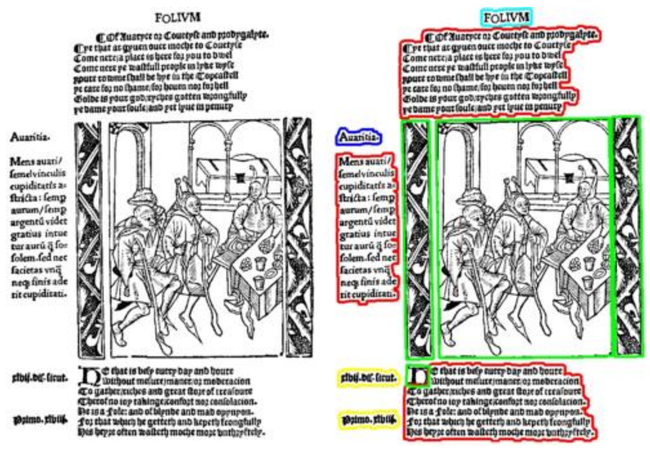
Weiterentwicklung eines semi-automatischen Open Source Tools zur Layout-Analyse und Regionen-Extraktion und -Klassifikation (LAREX) von frühen Buchdrucken

Lehrstuhl für Künstliche Intelligenz und Angewandte Informatik, Universität Würzburg
Kontakt: alexander.gehrke@uni-wuerzburg.de

Problem: Finden von Textregionen

- **Ziel:** Vorbereitung einer Buchseite für die Verarbeitung mit OCR durch Erkennen und Separieren der Textregionen und Bildregionen
- Eine Textregion ist ein Textblock mit klar definierter Lesereihenfolge
- Die Software soll einfache manuelle Korrektur der Regionen mit einem komfortablen User Interface bereitstellen
- Automatische Verfahren sollen manuelle Eingriffe maximal reduzieren.
- Weiteres Ziel: Regionen nach semantischer Bedeutung unterscheiden, z.B. Absätze / Überschriften oder Bilder / Initialen



Binarisierte Eingabeseite (links) und annotierte Regionsansicht in LAREX (rechts). Verschiedene semantische Regionstypen sind farblich markiert: Bilder und eine Initiale (grün), Fließtext (rot), Überschrift (blau), Marginalien (gelb) und Seitentitel (türkis).

LAREX

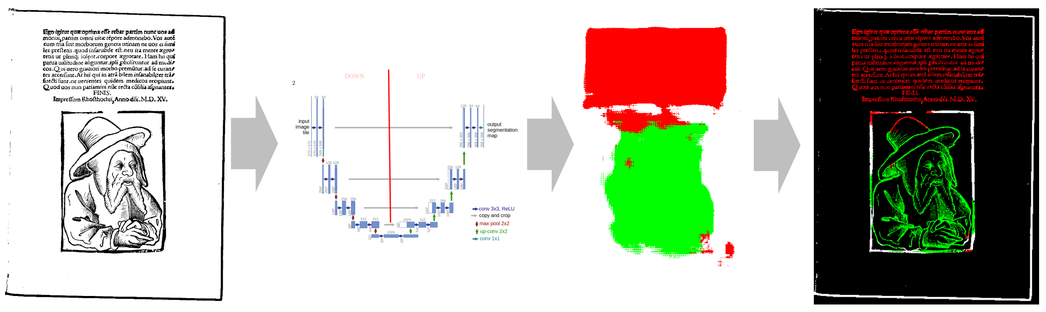
- Semi-automatisches Open-Source-Tool zur Layoutanalyse
- Enthält regelbasierten Connected-Components-Ansatz zur automatischen Regionenerkennung
- Nutzt PageXML als Ein- und Ausgabeformat zur Integration mit anderen Workflows
- Das Modulprojekt enthält Verbesserungen der Benutzeroberfläche, sowie die Entwicklung neuer automatischer Verfahren (noch nicht in LAREX integriert)
- In Benutzerstudien mit dem Narragonien Digital Projekt benötigte ein studentischer Assisten im Schnitt unter 4 Stunden zur Annotation eines Buchs mit ca. 300 Seiten (mithilfe der integrierten automatischen Verfahren).

Die automatische Erkennung konnte die Initiale und den Text nicht trennen, da sie graphisch miteinander verbunden sind.

Durch Verschieben der Regionsgrenze oder alternativ durch Ziehen eines Trennstriches oder Trennpolygons können fälschlicherweise zusammengewachsene Regionen manuell separtiert werden.

Pixel Classifier

- Zwei automatische Ansätze wurden basierend auf einem **Pixel Classifier** entwickelt
- Der Pixel Classifier nutzt ein **Fully Convolutional Neural Net (FCN)**, basierend auf der U-Net Architektur mit Skip Connections, um jedem Pixel der Eingabe eine semantische Klasse zuzuweisen.
- Das Eingabebild wird binarisiert und skaliert. Die Skalierung basiert auf der Buchstabenhöhe und muss während Training und Nutzung identisch sein.
- Das aktuelle Modell unterscheidet nur Text, Bild und Sonstiges (Rauschen, etc.), jedoch kann auch ein Modell mit mehr Klassen trainiert werden



Pixel Classifier Workflow: Ein Binärbild wird dem FCN übergeben (l.).
Das Netz gibt eine Klassifizierung für jedes Pixel aus, hier farbig visualisiert (2. v.r.).
Für die weitere Verarbeitung werden Hintergrundpixel ignoriert (r.)

Regionenextraktion

- Die Pixel werden durch Generierung von (approximativen) Rechtecken zu Blöcken (Regionen) zusammengefasst. Dazu werden Varianten des XY-Cut-Algorithmus verwendet.
  - **Classifier-first (CF)**: Dieser Ansatz klassifiziert zuerst die Pixel mittels des FCN.
    Anschließend werden die Vordergrundpixel jeder Klasse (typischerweise nur Text und Bild) separat mittels XY-Cut in Regionen aufgeteilt
  - **Segmentation-First (SF):** Der XY-Cut wird direkt auf das Binärbild angewandt. Die Regionen werden anschließend in Bild und Text klassifiziert.
- Die Ansätze erkennen unterschiedliche Regionen gut. Nur der CF-Ansatz kann Initialen erkennen, die im Absatz stehen, während der SF-Ansatz beim Trennen von Spalten bessere Ergebnisse liefert.

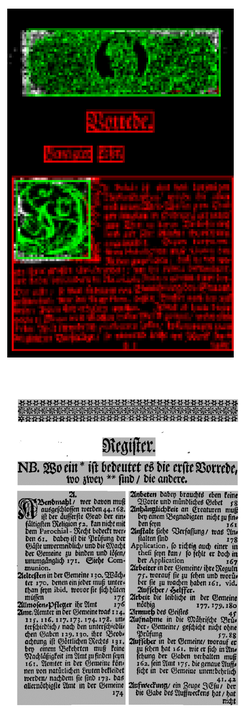


Abb. oben: Segmentierungsergebnis des CF-Ansatzes mit getrennter Initiale

unten: Spaltentrennung beim SF-Ansatz

Ergebnisse

Modelle:

- Kleines Modell: trainiert auf 108 Seiten aus 6 Werken, zusätzlich 14 Seiten zur Validierung. 3 Bild- oder Grafikregionen im PageXML.
- Großes Modell: trainiert auf 9137 Seiten aus 61 Werken, zusätzlich 266 Seiten zur Validierung. 611 Bild- oder Grafikregionen im PageXML.

Datasets:

- Trainingsdaten aus DTA 01 und 02, größtenteils 17. und 18. Jhdt.
- Inkonsistente Verwendung von <ImageRegion> und <GraphicsRegion>
- Überlappende TextRegion und GraphicsRegion in der GT
- Geringer Anteil an Seiten mit Bildern
- Mit mehr und besseren Ground-Truth-Daten sind Verbesserungen des Pixel Classifiers zu erwarten

Evaluation des Pixel Classifiers auf dem OCR-D Datensatz (173 Bilder), nur Vordergrundpixel:

	Kleines Modell			Großes Modell			Groß + CC Voting		
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Text	0.957	0.951	0.954	0.997	0.964	0.980	0.885	0.950	0.916
Bild	0	0	0	0.367	0.278	0.317	0.312	0.217	0.256
	Accuracy			Accuracy			Accuracy		
gesamt	0.914			0.934			0.836		

- Text wird größtenteils gut erkannt
  - Aber: schon kleine Fehler können zu Fehlern der OCR führen
- Bilder und Initialen werden oft nicht oder als Text erkannt
- Bei Connected Components Voting wird jeder CC eine Gesamtklasse nach Mehrheit der Pixelklassen zugewiesen. Dies verschlechtert jedoch die Ergebnisse

Evaluation der Regionenextraktion:

- Da die Regionenextraktion auf dem Pixelclassifier basiert, können Fehler bei diesem auch zu falschen Regionen führen. Beim Classifier-First-Ansatz (CF) beeinflusst dies die Regionenform, bei Segmentation-First (SF) nur die Kategorisierung.
- Die Regionenextraktion wurde manuell durch eine Fehleranalyse auf einem nach Inhaltstyp ausgewählten Datensatz durchgeführt:
  40 reine Textseiten, 30 Seiten mit Bildern, 15 mehrspaltige Seiten sowie 15 Titelseiten.

Ansatz	Gesamt	Nur Text	Mit Bild	Spalten	Titel
	Seiten mit kritischen Fehlern (nicht korrigierbar)				
CF	56.0%	25.0%	60.0%	86.7%	93.3%
SF	48%	17.5%	70.0%	66.7%	66.7%
	Seiten mit mindestens kleineren (korrigierbar, oder keine Auswirkung auf OCR) Fehlern				
CF	83.0%	67.5%	90.0%	93.3%	100%
SF	77.0%	62.5%	90.0%	73.3%	93.3%

- In vielen Fällen ist die vollautomatische Segmentierung nicht ausreichend
- Die verschiedenen Ansätze machen unterschiedliche Fehler, doch die Gesamtqualität unterscheidet sich wenig

Zusammenfassung

- Vollautomatische Regionenerkennung zeigt noch größere Fehler
- LAREX kann die manuelle Korrektur dieser stark beschleuningen
- Bessere GT kann die Qualität des FCN verbessern und die darauf aufbauenden Ansätze weniger fehleranfällig machen

Ausblick

- In LAREX durchgeführte Änderungen könnten direkt ins Modell einfließen.
- Manche Fehler des FCN könnten durch Postprocessing erkannt und behoben werden
- Ein andere vielversprechender Ansatz wird gerade erprobt. Er nutzt ein CNN um jede Connected Component statt jeden Pixel zu klassifizieren.

Referenzen

[1] Christian Reul, Uwe Springmann und Frank Puppe. „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 CulturalHeritage. 2017, S. 137–142

[2] Christoph Wick und Frank Puppe. „Fully convolutional neural networks for pagesegmentation of historical document images“. In:2018 13th IAPR InternationalWorkshop on Document Analysis Systems (DAS). IEEE. 2018, S. 287–292.