



PhD-FSTC-2018-58
Fakultät für Naturwissenschaften, Technologie und Kommunikation

DISSERTATION

verteidigt am 17/09/2018 in Luxemburg

zur Erlangung des Titels

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG EN INFORMATIQUE

von

Winfried Höhn

geboren am 25. Juni 1981 in Dettelbach (Deutschland)

SEMIAUTOMATISCHE ANNOTATION VON ORTEN IN DIGITALISIERTEN ALTKARTEN

Prüfungskommission

Dr. Steffen Rothkugel, Vorsitzender
Associate Professor, Universität Luxemburg

Dr. Alexander Wolff, Stellvertretender Vorsitzender
Professor, Universität Würzburg

Dr. Christoph Schommer, Betreuer der Doktorarbeit
Associate Professor, Universität Luxemburg

Dr. Andreas Fickers
Professor, Universität Luxemburg

Dr. Christof Schöch
Professor, Universität Trier

Dr. Karsten Tolle, Beratender Experte
Professor, Universität Frankfurt

Zusammenfassung

Altkarten werden zunehmend in die Bestände digitaler Bibliotheken aufgenommen, um den Zugang zu diesen historisch wertvollen Dokumenten einer breiteren Masse an Benutzern zu ermöglichen. Dabei wurde der Bedarf identifiziert, die digitalen Kopien der Altkarten mit Annotationen anzureichern, die eine schnelle Suche nach relevanten Informationen und Verlinkung mit anderen Dokumenten ermöglichen. Da eine solche manuelle Annotation von Altkarten mit einem hohen Arbeitsaufwand und hohen Kosten verbunden ist und deswegen nur für besonders wertvolle Dokumente vorgenommen wird, bleibt die große Masse der Altkarten unerschlossen. Mit dieser Motivation wurde in der vorliegenden Forschungsarbeit das Ziel gesetzt, die in den Altkarten enthaltenen Informationen weitgehend automatisch zu finden und zu annotieren.

Allerdings sind die existierenden Methoden zur automatischen Informationsextraktion aus Landkarten auf moderne Karten ausgelegt und beschränken sich meist auf bestimmte Kartentypen, die ähnliche grafische Merkmale aufweisen. Die Annotation von Altkarten wurde bisher nur durch Werkzeuge unterstützt, die eine manuelle Erfassung der Kartendaten und eine digitale Speicherung der annotierten Informationen ermöglichen.

Ausgehend von diesem zu Beginn dieser Forschungsarbeit gegebenen Rahmen wurden Forschungsfragen formuliert, die auf eine automatische Identifikation der Ortssignaturen, automatische Identifikation Textlabels und deren automatische Texterkennung (*Optical Character Recognition, OCR*) und eine automatische Verlinkung mit einer modernen Karte oder einer geographischen Datenbank ausgerichtet sind. Um die gestellten Forschungsfragen zu beantworten wurden hauptsächlich Methoden des maschinellen Lernens genutzt. Sowohl klassische Methoden des maschinellen Lernens als auch Deep-Learning-basierte Verfahren wurden erweitert, um den Anforderungen der spezifischen Domäne der Altkarten gerecht zu werden. Als Datensatz wurden Altkarten von Luxemburg und dem Fränkischen Reichskreis benutzt.

Um Ortssignaturen automatisch zu finden, wurde in dieser Arbeit eine Template-Matching-Variante entwickelt, die robust gegenüber Störobjekten in der unmittel-

telbaren Nähe der zu erkennenden Ortssignaturen ist. Die Methode ist auch in anderen Bereichen mit ähnlichen Eigenschaften anwendbar. Weiterhin wurden sowohl regelbasierte als auch statistische Verfahren auf die Domäne der Altkarten zugeschnitten, um die Position der Textlabels in den Altkarten zu finden. Um die identifizierten Texte lesbar zu machen, wurde eine eigene OCR-Engine implementiert, die nur einem Viertel der Zeichenfehler aufweist, die ein übliches OCR-Tool, bei Verwendung derselben Trainingsdaten, erreicht. Die Georeferenzierung wurde sowohl zwischen einer Altkarte und einer modernen Karte, als auch zwischen zwei Altkarten untersucht. Es wurde festgestellt, dass eine automatische Georeferenzierung zwischen zwei Altkarten nicht nur genutzt werden kann, um Fehler in der manuellen Annotation zu finden, sondern auch neue historische Erkenntnisse liefern kann, wie beispielsweise kleine Änderungen in den Kupferplatten über die Zeit, von denen die Karten gedruckt wurden (z.B. Hinzufügen von einer kleinen Anzahl von Orten, das sonst nirgendwo dokumentiert ist). Zum Erfassen von Trainings- und Testdaten wurde ein Referenzierungs- und Annotationstool implementiert. Dieses kann auch genutzt werden um die Automatisierungen zu testen sowie automatisch erfasste Daten zu korrigieren.

Insgesamt leistet die vorliegende Arbeit einen Beitrag in den Bereichen Objekterkennung in Grafiken, Scene Text Detection, OCR und Georeferenzierung. Es wird gezeigt, mit welchem Aufwand einzelne Teile und der Gesamtbereich der Altkarten automatisch lesbar gemacht werden kann und wo die momentanen Grenzen der Automatisierung liegen. Zum Schluss zeigt diese Arbeit, wie groß das Potential und die Wirkung solcher digitalen Werkzeuge im ganzen Bereich *Digital Humanities* ist.

Abstract

Old maps are increasingly being digitised and included in digital libraries to provide access to these historically valuable documents to a wider audience. The need was identified to enrich the digital copies of the old maps with annotations that enable a quick search for relevant information and linking with other documents. Because such a manual annotation of old maps is a labour- and cost-intensive process and is therefore only carried out for particularly valuable documents, the large mass

of old maps remains unexplored. With this motivation, the aim of this research project was to automatise to a great extent the process of object detection and annotation in the old maps.

However, existing methods for automated information extraction from maps are designed for modern maps. They are usually limited to certain map types that have similar graphical characteristics. Alternatively there are tools for manual annotation of map data. Until now, annotation of old maps was only supported by tools that enabled manual extraction of map data and the digital storage of annotated information.

The research objectives for this dissertation have been formulated based on the research frame given at the beginning of this work. The research objectives include automatic identification of place markers, automatic detection of text labels, their Optical Character Recognition (OCR) and an automatic linking with a modern map or a geographical database. In order to meet the research objectives, mainly methods of machine learning were used. Both classic machine learning methods and deep-learning-based methods have been extended to meet the requirements of the specific domain of old maps. Old maps of Luxembourg and the Franconian Circle were used as a data set.

In order to find place markers automatically, a template matching variant was developed which is robust against interfering objects next to the place markers to be recognised. The method is also applicable in other areas with similar properties. Furthermore, both rule-based and statistical methods were tailored to the domain of the old maps with the purpose to find the position of the text labels. In order to make the identified texts readable, a new OCR engine was implemented that produces only a quarter of the character errors compared with a state-of-the-art OCR tool, while using the same training data. The georeferencing was examined for two types of map pairs: an old and a modern map, and two old maps. It was found that automatic georeferencing between two old maps can not only be used to find errors in manual annotations, but can also provide new historical insights, such as small changes in the copper plates from which the maps were printed (e.g. adding a small number of places that is not documented anywhere else). A referencing and annotation tool has been implemented to collect training and test data. This tool can also be used to test the automations and to correct

automatically collected data.

Overall, the present work makes a contribution in the areas of object recognition in graphics, scene text detection, OCR and georeferencing. It is shown which effort is needed to make individual parts or the entire area of old maps automatically readable and where the current limits of the automation are. Finally, this research shows the huge potential and the impact of such digital tools in the entire area of digital humanities.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation und Problemstellung	1
1.2. Forschungsfragen und Ziele der Arbeit	4
1.3. Beitrag und Aufbau der Arbeit	7
2. Stand der Forschung	9
2.1. Erkennung von Ortssignaturen	10
2.2. Automatische Erkennung von Text in Bildern	12
2.2.1. Techniken zur Textlokalisierung in Bildern	13
2.2.2. Techniken zur automatischen Texterkennung	16
2.3. Zusammenführen ähnlicher Karten	18
2.4. Georeferenzierungs- und Annotationstools für Karten	21
3. Erkennung von Ortssignaturen	23
3.1. Altkarten als Problemfeld für die automatische Extraktion von Kartensymbolen	24
3.2. Template-Matching-basierter Ansatz	27
3.3. Deep-Learning-basierter Ansatz	32
3.4. Experimente	35
3.4.1. Experimentaufbau	35
3.4.2. Evaluierungsmethodik	36
3.4.3. Auswirkung der Templategröße	38
3.4.4. Ergebnisse	39
3.4.5. Diskussion der Ergebnisse	57
3.5. Zusammenfassung	58

4. Erkennung der Beschriftung von Ortssignaturen	59
4.1. Erkennung von Textbereichen	59
4.1.1. Modifiziertes regelbasiertes Verfahren zur Textlokalisierung in Altkarten	61
4.1.2. Deep Learning zur Textlokalisierung in Altkarten	67
4.1.3. Evaluierung	70
4.2. Texterkennung	76
4.2.1. Ocropy	76
4.2.2. Training eines eigenen Modells	85
4.2.3. Evaluierung	89
4.3. Diskussion der Ergebnisse	92
4.4. Zusammenfassung	94
5. Verknüpfung mit einer geographischen Datenbank	97
5.1. Das Problem der Georeferenzierung von Ortsmarkern in Altkarten	98
5.2. Heuristiken zur Verlinkung von Ortsmarkern mit einer geographi- schen Datenbank	99
5.3. Evaluierung	102
5.3.1. Evaluierungsmetriken und Experimente	103
5.3.2. Diskussion der Ergebnisse	106
5.4. Zusammenfassung	106
6. Kartenübergreifende Annotationsübertragung	107
6.1. Identifikation korrespondierenden Ortssignaturen	108
6.2. Evaluierung	113
6.2.1. Experimentaufbau	113
6.2.2. Diskussion	125
6.3. Zusammenfassung	126
7. RAT - Referenzierungs- und Annotations-Tool	127
7.1. Ortsdaten	127
7.2. Benutzeroberfläche	129
7.3. Zusammenfassung	135
8. Fazit	137
9. Ausblick	145

A. Karten	149
A.1. Luxemburg	149
A.1.1. Lutzenburgensis Ducatus / Pieter van den Keere (1617) . .	149
A.1.2. Lutzenburgensis Ducatus / Johannes Janssonius van Waes- berge (1636)	150
A.1.3. Carte du Duché de Luxembourg / Didier Robert de Vau- gondy (1753)	152
A.1.4. Lutzembvrgensis / Abraham Ortelius (1608)	153
A.1.5. Lutzembvrgensis Ducatus Veriss. Descript. / Abraham Or- telius (1579–1612)	155
A.2. Fränkischer Reichskreis	156
A.2.1. Das FranckenLandt = Chorographi Franciae Orien[talis] / Sebastian von Rotenhan (1533)	159
A.2.2. Franciae orientalis (vulgo Franckenlant) descriptio / Sebas- tian von Rotenhan/Abraham Ortelius (nach 1570)	160
A.2.3. Franckenlandt = Francia orientalis / Gerhard Mercator (1600)	162
A.2.4. Nova Franconiae descriptio / Abraham Goos (1626)	163
A.2.5. Franconia : Newe Carten Des Gantzen Fränckischen Krayses / Matthäus Merian (1656)	164
A.2.6. Circulus Franconicus / Matthias Seutter (1731)	166
A.2.7. Das Fürstenthum Würzburg / Fackenhofen (1791/1805) . .	168
A.2.8. Franconia vulgo Franckenlandt / Willem Janszoon Blaeu (1650)	170
A.2.9. Carte Topographique D’Allemagne / Daniel Adam Hauer (1787)	171
A.2.10. Circulus Franconicus / Matthaeus Seutter (1731)	172
Literaturverzeichnis	175

1. Einleitung

Altkarten und historische Drucke bieten einen einzigartigen Einblick in die Vergangenheit und sind daher von unermesslicher wissenschaftlicher und kultureller Bedeutung. Digitale Sammlungen von unterschiedlichen historischen Dokumenten werden zu einem notwendigen und unersetzlichen Werkzeug, um Wissenschaftler, Historiker und Bibliothekare in ihren Forschungsaufgaben im digitalen Zeitalter zu unterstützen. Diese und andere potentielle Nutzergruppen, wie jede historisch interessierte Person, benötigen technologische Unterstützung, um spezifische Karten oder relevante Stellen in diesen Karten zu finden.

Die Retrodigitalisierung von Altkarten erzeugt deren digitale Versionen, die an sich nur Rasterbilder sind. Häufig werden die Karten danach mit Metainformationen wie Autor, Kartentitel, Entstehungsjahr usw. versehen. Dabei werden aber nicht die in den Karten enthaltenen geographischen Informationen erschlossen. Die Anreicherung mit Metadaten ist Kernvoraussetzung, um nach Dokumenten suchen zu können und diese zu verknüpfen. Der Digitalisierungsprozess sollte daher nicht mit der Erstellung des Digitalbildes enden, sondern sollte auch die Erfassung der Semantik beinhalten.

1.1. Motivation und Problemstellung

Eine über die Metadaten hinausgehende Erschließung der eingescannten Altkarten muss meist in Handarbeit erfolgen und wird wegen des hohen Aufwandes nur für besondere Werke vorgenommen. Um eine möglichst vollständige semantische Annotation für alle historischen Dokumente einer Sammlung realisieren zu können, muss der Aufwand für die Erschließung der in den Dokumenten enthaltenen

1. Einleitung

Informationen reduziert werden. Dies kann unter anderem durch das Bereitstellen von geeigneten Werkzeugen und einer möglichst weitreichenden Automatisierung der Annotation geschehen. Die technologische Unterstützung kann sich von einer computerunterstützten Annotation bis zu einer vollständig automatisierten Analyse von Karten erstrecken.

Viele Bibliotheken und Archive, die große Altkartensammlungen besitzen, stellen Teile dieser Sammlungen auch online zur Verfügung. Meist werden die Karten auf einem eigenen Portal mit einigen Metadaten zur Karte aber ohne deren inhaltliche Erschließung präsentiert. Einige Beispiele hierfür sind:

University Library Bern – Map collections Die Sammlung von historischen Karten der Universitätsbibliothek Bern zählt zu einer der umfangreichsten weltweit und ist auch in großen Teilen digitalisiert und online zugänglich. Alleine die Kartensammlung *Ryhiner* besteht aus ca. 16.000 Landkarten, Plänen und Ansichten, welche vom Berner Staatsmann Johann Friedrich von Ryhiner im 18. Jahrhundert zusammengetragen wurde und Karten der ganzen Welt enthält. Hinzu kommen noch umfangreiche Bestände an topographischen Karten der Schweiz. (UB Bern Kartensammlung, 2018; UB Bern Rhyner, 2018)

Sächsische Landesbibliothek – Staats- und Universitätsbibliothek Dresden (SLUB) – Virtuelles Kartenforum Ziel des Virtuellen Kartenforums ist es Werkzeuge zur Georeferenzierung von Altkarten hauptsächlich des Deutschen Reiches bereitzustellen. Im Kartenforum sind online momentan 4095 Karten verfügbar, welche aus den Zeiten von 1625 bis 1965 kommen. Dabei stammen 513 dieser Karten aus der Zeit vor 1850. (Virtuelles Kartenforum Projekt, 2018; Virtuelles Kartenforum, 2018)

David Rumsey Map Collection David Rumsey hat vor über 30 Jahren damit begonnen seine Kartensammlung, welche über 150.000 Karten umfasst, aufzubauen. Der Schwerpunkt der Sammlung liegt auf Karten von Nord- und Südamerika vom 16. Jahrhundert bis heute. Die Sammlung enthält aber auch umfangreiche Bestände an Karten aus der ganzen Welt. Die komplette Sammlung wurde von David Rumsey der Stanford Universität

1.1. Motivation und Problemstellung

geschenkt, wo sie besichtigt werden kann und auch die digitale Sammlung archiviert wird. Die Digitalisierung begann 1996 und mittlerweile sind über 86.000 Objekte der Sammlung digitalisiert. Die Digitalisate sind online frei zugänglich. (Rumsey Historical Map Collection, 2018)

Franconica-Online Das Projekt Topographia-Franconiae hat sich das Ziel gesetzt, historische geografische Ortsinformation vom frühen 16. Jahrhundert bis ins frühe 20. Jahrhundert aus dem ehemaligen Fränkischen Reichskreis zu erschließen. (Schmidt, 2007)

Ein Meta-Portal, welches eine Suche über mehrere unabhängige Kartensammlungen ermöglicht, ist das OldMapsOnline-Portal, welches im Januar 2018 36 Online-Kartensammlungen (OldMapsOnline, 2018) auflistet, die mit der Suchoberfläche des Portals gemeinsam durchsucht werden können.

Auch Google Maps bot mit dem *Google Maps Engine public data program* die Möglichkeit, öffentliche Kartendaten über die Google Infrastruktur auszuliefern und mit Google Maps zu verknüpfen. Diese Möglichkeit wurde z. B. von National Geographic genutzt, um mehr als 500 Karten aus der über hundertjährigen Geschichte des Magazins für die Welt öffentlich zugänglich zu machen (Biasi, 2013; Howard, 2014). Mit Einstellung der *Google Maps Engine* im Januar 2016 wurden diese Daten aber alle unzugänglich (Google Maps Engine, 2016).

In solchen Sammlungen einzelne digitalisierte Dokumente aufzufinden und nach den darin enthaltenen Informationen zu suchen stellt ein Problem dar. Meist sind nur Metadaten über die Dokumente, nicht aber die Dokumentinhalte erschlossen. So können auf OldMapsOnline Karten nach ihren Metadaten gefunden werden, wie beispielsweise dem Kartenausschnitt, Titel oder Autor. Die eigentlichen Karten können aber nicht nach den abgebildeten geografischen Informationen durchsucht werden. Für eine leichtere Auffindbarkeit ist ein höherer Grad der Erschließung und Annotation von Dokumentinhalten nötig.

Altkarten, wie auch Karten allgemein, beinhalten diverse Typen von Informationen, wie zum Beispiel Orte, Flüsse, Straßen, Berge, Wälder und deren Positionen. Speziell für Ortssymbole können Informationen wie der Ortstyp, Ortsname, Geo-

1. Einleitung

koordinaten und Korrespondenz zu einem modernen Ort erfasst werden und für Fragestellungen in angrenzenden Forschungsgebieten nützlich sein. Die existierenden Tools helfen den für die Kartendarstellung genutzten Bereich zu erfassen und auch die Geokoordinaten für ausgewählte Punkte der Karte zu definieren. Dies ermöglicht es, den von der Kartendarstellung abgedeckten Bereich abzuschätzen und nach Karten für bestimmte geographische Bereiche zu suchen. Da dies ein aufwändiger Prozess ist, wird dies meist durch Crowdsourcing bewältigt. *Georeferencer* (Fleet et al., 2012) und *Recogito* (Simon et al., 2015, 2017) repräsentieren den heutigen Entwicklungsstand auf diesem Gebiet. Sowohl in *Georeferencer* als auch in *Recogito* ist es möglich, Texte in Karten von Hand zu erfassen, z.B. Ortsnamen. Dies zeigt, dass auch ein Interesse besteht diese Informationen aus Karten zu extrahieren. Diese Tools bieten aber nicht die Möglichkeit solche Informationen systematisch und automatisiert zu erfassen und mit den Ortssignaturen zu verknüpfen.

Das Problem der automatischen Erkennung von Ortssignaturen kann als Spezialfall der Objekterkennung im Forschungsgebiet des maschinellen Sehens aufgefasst werden. Der Fokus der Objekterkennung im Allgemeinen liegt auf dem Identifizieren bekannter Objekte in einem Bild. Der momentan sehr aktive Bereich der *Scene Text Detection/Recognition* befasst sich mit der Erkennung von Textbereichen und dem Erkennen des Textes in Bildern und Fotos, was mit der Erkennung von Textlabels auf einer Karte und dem Erkennen des Texts dieser Labels verwandt ist. Es muss jedoch noch weiterer Aufwand betrieben werden um die vorhandenen Techniken auch im Bereich der Altkarten anwenden zu können. Auf die Punkte, die eine Anwendung im Bereich der Altkarten erschweren und wie dies gelöst werden kann, wird in den jeweiligen Abschnitten eingegangen.

1.2. Forschungsfragen und Ziele der Arbeit

Zunächst soll noch einmal betont werden, dass das Kartenmaterial sehr unterschiedlich ist. Moderne Karten, die in vielen Forschungsarbeiten im Fokus stehen, haben grundsätzlich andere Charakteristiken als die Altkarten, deren automatische Erschließung im Mittelpunkt dieser Arbeit steht. Moderne Karten sind in hohem Maße standardisiert. Für die Verwendung von Farbkodierung, Schrift-

1.2. Forschungsfragen und Ziele der Arbeit

arten, Schriftgrößen, Symboltypen gibt es klare Vorgaben, die innerhalb einer Karte strikt eingehalten werden. Moderne Technik erlaubt es, diese Standards sehr genau einzuhalten, beispielsweise durch topographische Software und moderne Druckverfahren. Diese technische Unterstützung war zum Zeitpunkt der Erstellung der Altkarten nicht denkbar. Die Linien sind daher nicht so fein, die Symbole mit derselben Semantik sind oft ähnlich, aber nicht identisch, und können daher maschinell nicht immer korrekt erkannt werden, weil bei der manuellen Herstellung viele Ungenauigkeiten aufgetreten sind. Die Schöpfer der Altkarten haben sich außerdem viele künstlerische Freiheiten erlaubt, beispielsweise in der Variation der Ortssignaturen, der Symbole, welche die Orte auf der Karte repräsentieren.

Der vorhergehende Abschnitt 1.1 beschreibt unterschiedliche Möglichkeiten und Schwerpunkte für die Annotation historischer Dokumente. Im Bereich der Altkarten kann bereits vieles erreicht werden, wenn man die Ortssignaturen und die dazugehörigen Ortsnamen automatisch extrahieren kann, da die Orte und deren Namen einen Großteil der auf der Karte enthaltenen Informationen bilden. Eine Verknüpfung der gefundenen Orten und deren Namen mit einer modernen Karte oder einer geografischen Datenbank ermöglicht dann auch eine Suche innerhalb von Karten. Dadurch wird beispielsweise eine Suche nach historischen Dokumenten, die einen Ort betreffen, möglich. Eine Verknüpfung der verschiedenen Dokumente über die gerade betrachteten Orte wäre damit auch machbar. Durch die Verknüpfung mit einer modernen Karte sind regionale Suchen möglich. Die Geodatenbank kann mit Zusatzinformationen erweitert werden um z. B. auch nach historischen Namensvarianten suchen zu können. Diese Zusammenhänge sind in Abbildung 1.1 dargestellt.

Das Ziel dieser Arbeit ist es, den Aufwand für die Erfassung aller in einer Altkarte dargestellten Orte anhand ihrer Ortssignaturen, deren Beschriftungen und der Zuordnung zueinander weitgehend zu automatisieren. Die speziellen Forschungsfragen beziehen sich auf die Erfassung folgender Typen von Metadaten: die Position und den Typ der Ortssignatur, deren Beschriftung und eine Verknüpfung zu den modernen Ortsdaten. Die Forschungsfragen sind:

1. Lassen sich Ortssignaturen in Altkarten automatisch erkennen?
2. Wie lässt sich die Beschriftung der Ortssignaturen automatisiert erfassen im

1. Einleitung

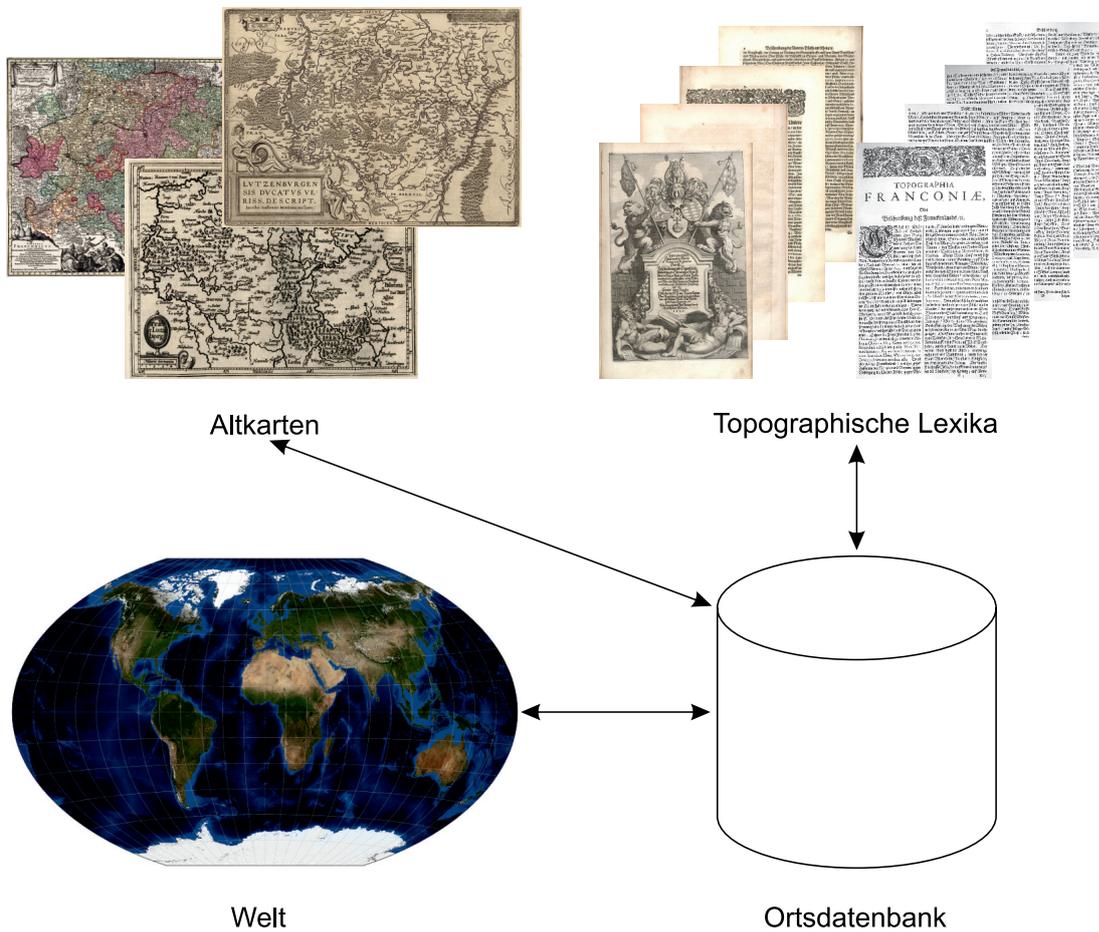


Abb. 1.1.: Verknüpfung der annotierten Dokumente über eine Ortsdatenbank mit den in den Dokumenten enthaltenen Orten

Sinne von Label-Identifikation und Label-Texterkennung?

3. Wie lässt sich die Zuordnung zu modernen Orten erleichtern oder automatisieren (Georeferenzierung)?
4. Wie lässt sich die Ähnlichkeit zwischen Karten nutzen um die Erfassung der Orte zu unterstützen?

Um diese Forschungsfragen zu beantworten, werden hauptsächlich Techniken des maschinellen Lernens aus dem Bereich der Bilderkennung eingesetzt und weiterentwickelt, die sowohl klassisches Template-Matching als auch Verfahren aus

dem Bereich des Deep-Learning einschließen. Die anwendungsbezogenen Resultate werden in einem Referenzierungs- und Annotationstool (RAT) implementiert.

1.3. Beitrag und Aufbau der Arbeit

Diese Arbeit hat einen wesentlichen Beitrag im Bereich Objekterkennung in Grafiken geleistet, indem eine Template-Matching-Variante entwickelt wurde, die robust gegenüber Störobjekten in der unmittelbaren Nähe der zu erkennenden Objekte ist. Diese Methode ist auch auf andere Domänen mit ähnlichen Eigenschaften (zum Beispiel technische Zeichnungen mit vielen kleinen Symbolen) übertragbar.

Zusätzlich leistet diese Dissertation einen praktischen Beitrag im Bereich Scene Text Detection, indem regelbasierte Methoden und Verfahren des maschinellen Lernens auf die herausforderungsvolle Domäne der Altkarten zugeschnitten werden. Die resultierende Software löst nicht nur ein wissenschaftliches Problem der Textlabelerkennung in manuell erstellten Karten, sondern bietet auch eine funktionierende Lösung für den Endbenutzer.

Darüberhinaus leistet diese Arbeit einen deutlichen Beitrag im Bereich *Optical Character Recognition* (OCR), indem eine neuartige OCR-Engine implementiert wurde, die mit wenigen Trainingsdaten auskommt. Um dies zu ermöglichen, musste eine neue Modellarchitektur gefunden werden.

Außerdem leistet diese Dissertation einen Beitrag im Bereich Georeferenzierung, indem ein innovatives Vorschlagssystem entwickelt wurde, welches Ortskandidaten mit hoher Genauigkeit vorschlägt, und somit den kognitiven Aufwand der manuellen Verlinkung zwischen einer Altkarte und einer modernen Karte minimiert. Zum Schluss wurde in dieser Arbeit eine automatische Identifikation ähnlicher Altkarten implementiert, die zum einen als Georeferenzierung zwischen zwei Altkarten dient, und zum anderen für eine einfachere Verlinkung der Ortsdaten zu einer modernen Karte genutzt werden kann.

Insgesamt hat diese Arbeit gezeigt, mit welchem Aufwand einzelne Teile und der Gesamtbereich der Altkarten automatisch erschlossen werden kann, um die Alt-

1. Einleitung

karten für Suchmaschinen lesbar und somit für unterschiedliche Benutzergruppen erst sinnvoll benutzbar zu machen.

Die einzelnen Bereiche werden in dieser Dissertation wie folgt beschrieben:

Kapitel 2 diskutiert die existierenden Arbeiten in den relevanten Bereichen wie Feature-Extraktion, automatische Identifikation von Kartensymbolen, Erkennung von Text sowie Möglichkeiten zur Verknüpfung mit einer geografischen Datenbank beschrieben. Außerdem wird die Anwendbarkeit der genannten Methoden auf die für diese Arbeit zur Verfügung stehenden Daten beschrieben und die Nachteile der existierenden Verfahren analysiert, um den eigenen Beitrag zur Forschung deutlich zu machen.

Kapitel 3 stellt das neue Verfahren zur automatischen Erfassung der Kartensymbole vor.

Kapitel 4 erklärt die in dieser Dissertation gewählte Vorgehensweise um Textlabels in Karten zu finden und den Text zu erkennen.

Kapitel 5 zeigt die Möglichkeiten zur Verknüpfung der gewonnenen Informationen mit einer modernen geografischen Datenbank auf. Der eigene Lösungsansatz wird vorgestellt.

Kapitel 6 beschreibt einen Weg, den Annotationsaufwand durch die Identifizierung ähnlicher Karten und die Ausnutzung von Informationen in diesen zu reduzieren.

Kapitel 7 stellt das Referenzierungs- und Annotationstool RAT vor, in dem die neuen Ansätze implementiert wurden.

Kapitel 8 fasst die Ergebnisse der vorliegenden Arbeit zusammen und diskutiert potentielle Erweiterungen, die auf dem vorgestellten Ansatz basieren können, sowie dessen technische Grenzen.

Eine Beschreibung der für diese Arbeit verwendeten Karte befindet sich im Anhang A.

2. Stand der Forschung

Das Erkennen und Annotieren von Objekten in Bildern ist in den letzten Jahrzehnten zu einem sehr aktiven Forschungsfeld im Bereich der Mustererkennung und Computer Vision geworden. Es erstreckt sich von der Gesichtserkennung über die Segmentierung von Bildern in Objekte und das Zuweisen von Stichworten zu Bildern bis hin zur Text- und Handschriftenerkennung.

Durch die gewonnenen Informationen wird eine Suche innerhalb von Bildersammlungen erst möglich. So bietet beispielsweise Google eine Suche nach ähnlichen Bildern an sowie die Möglichkeit, die Typen der angezeigten Bilder auf Gesichter, Fotos oder Strichzeichnungen zu beschränken.

In einem frühen Stadium existieren Techniken, um Wortvorkommen anhand eines Beispielbildes des Wortes in historischen Dokumenten zu finden (Rath und Manmatha, 2007; Rusiñol et al., 2015; Sudholt und Fink, 2016). Diese Bildannotationen und die damit erweiterten Suchmöglichkeiten sind für Digitale Bibliotheken von großer Bedeutung, besonders wenn sie sich mit historischen Dokumenten, wie z. B. Landkarten, Drucken oder Urkunden, befassen. Die zu annotierenden Objekte sind je nach Dokumenttyp unterschiedlich. So sind es in den Landkarten unter anderem Orte, Straßen und Flüsse und für Drucke und Handschriften in erster Linie der Text. Annotation und Georeferenzierung machen die in den Altkarten und historischen Dokumenten enthaltenen Informationen erst für Suchmaschinen und Information-Retrieval-Systeme erschließbar. Dadurch werden Referenzen zwischen unterschiedlichen Informationsquellen ermöglicht, wie zum Beispiel Altkarten und moderne Karten für dieselbe Region, oder historische topographische Lexika.

Unterschiedliche Verfahren zur automatischen Erkennung von diversen Typen von Karteninhalten sind hauptsächlich für moderne Karten entworfen worden. Moder-

2. *Stand der Forschung*

nes Kartenmaterial verfügt über ein hohes Maß an Regelmäßigkeit. Beispielsweise werden Farben gezielt eingesetzt, um bestimmte Informationen zu kodieren, die Linienstärke ist bei jedem Linientyp (fast) durchgehend gleich, Ortsmarkierungen sind normiert und typisiert. Nur wenige von den Verfahren, die für solche Daten geschaffen wurden, lassen sich auf manuell und ohne jegliche Normierung erstellte Altkarten übertragen. In diesem Kapitel wird ein ausführlichen Überblick über existierende und vor allen Dingen auf Altkarten anwendbare Methoden gegeben, welche zur automatischen Feature-Extraktion, automatischen Identifikation von Kartensymbolen sowie Erkennung der Beschriftung verwendet werden können. Darüber hinaus werden relevante Arbeiten für moderne Karten analysiert. Zum Schluss werden Möglichkeiten zur Verlinkung zwischen einer modernen Karte bzw. einer geographischen Datenbank diskutiert und bereits in diesem Gebiet existierende Tools beschrieben.

2.1. **Erkennung von Ortssignaturen**

Das Problem der Extraktion der Ortssymbole in den Altkarten kann als eine Instanz des Template-Matchings oder der Objekterkennung formuliert werden. Template-Matching-Verfahren sind eine etablierte Methode aus der Bildverarbeitung und werden zur Symbolerkennung in vielen Bereichen eingesetzt. Eine ausführliche Beschreibung der Einsatzbereiche und anwendbaren Methoden für Schaltpläne, Konstruktionszeichnungen, Karten, Partituren, Grundrissplänen, Logo Erkennung und weitere Anwendungsbereichen findet sich in Lladós et al. (2002). Die Karten werden bei Lladós et al. (2002) in Katasterpläne, Spartenpläne und Landkarten unterteilt, wobei Landkarten als der komplizierteste Teilbereich beschrieben werden. Generell werden Vorkommen eines Musterbildes (Template) im Bild gesucht. So kann eine in der Legende der Altkarte oder in der Karte selbst enthaltene Ortssignatur als Template gesehen werden und alle dazu passenden Stellen im Kartenbild mit Hilfe des Template-Matching-Verfahrens gefunden werden.

Für das Template-Matching kommen beispielsweise die Summe der quadrierten Differenzen oder der Korrelationskoeffizient sowie deren normalisierte Form als Abstandsfunktion in Frage (OpenCV Template Matching, 2018). Die normalisierte Kreuzkorrelation (NCC) ist invariant gegenüber Helligkeits- und Kontrastände-

2.1. Erkennung von Ortssignaturen

rungen. Änderungen wie Skalierung oder Rotation führen aber dazu, dass das Template nicht gefunden werden kann. Chiang et al. (2014) führten die normalisierte Kreuzkorrelation als eine Möglichkeit auf um Templates von Straßenkreuzungen in einer Karte zu finden.

Unter Objekterkennung werden alle Techniken zusammengefasst, die Instanzen von Objekten in Bildern lokalisieren und identifizieren können. Durch die Verwendung von Convolutional Neural Networks (CNNs) konnten in den letzten Jahren große Fortschritte auf diesem Gebiet erzielt werden (Huang et al., 2017). Die momentane Dominanz der CNNs begann 2012 mit dem Sieg von AlexNet (Krizhevsky et al., 2012) bei der *ImageNet Classification Challenge*, in welcher für ein Bild das darin zu erkennende dominante Objekt aus 1.000 Kategorien ausgewählt werden muss. Die Deep-Learning-Objekterkennung muss zusätzlich zur Klassifikation noch die Bounding-Boxen der Objekte bestimmen. Hierzu wird eine Meta-Architektur verwendet, die aus einem Merkmalsextraktor, einem CNN wie z. B. AlexNet, besteht und bestimmt, wie damit die Bounding-Boxen erkannt und für diese anschließend die Klassifikation durchgeführt wird. Die ersten Architekturen, wie R-CNN (Girshick et al., 2013), bestanden aus einem Objektdetektor, welcher Bounding-Boxen für Objekte unabhängig von ihrer Klasse erstellt und welchen in einem zweiten Schritt die entsprechende Klasse zugeordnet wird. Neuere Architekturen wie YOLO (Redmon et al., 2016) und SSD (Liu et al., 2015) kombinieren diese beiden Schritte, was die benötigte Rechenleistung reduziert. Die Meta-Architekturen für die Objekterkennung können frei mit Merkmalsextraktoren, wie z.B. AlexNet (Krizhevsky et al., 2012), VGG (Simonyan und Zisserman, 2014), InceptionV3 (Szegedy et al., 2016) oder ResNet (He et al., 2015), kombiniert werden.

Die Deep-Learning-basierten Verfahren weisen eine sehr große Robustheit gegenüber Änderungen auf, benötigen aber eine immense Menge an Trainingsdaten. Dahingegen sind die Template-Matching-Verfahren nur robust gegenüber Änderungen, welche in ihrer Implementierung vorgesehen sind wie Helligkeits- und Kontraständerungen bei der normalisierten Kreuzkorrelation.

2.2. Automatische Erkennung von Text in Bildern

Die automatische Erkennung von Text in Bildern kann in zwei abstrakte Stufen eingeteilt werden:

1. Automatische Erkennung von Textbereichen (engl.: *text-graphic separation*)
2. Automatisches Einlesen des Textes (engl.: *optical character recognition (OCR)*).

Der deutsche Begriff *Texterkennung* wird häufig als Synonym für beide Teile dieses Prozesses verwendet und kann somit zu Unklarheiten führen. Um dies zu vermeiden, wird in dieser Arbeit der Begriff *Erkennung von Textbereichen* oder *Textlokalisierung* für den ersten Schritt in diesem Prozess verwendet. Der Begriff *Texterkennung* oder *OCR* wird verwendet um den zweiten Schritt dieses Prozesses zu referenzieren.

Eine Trennung in diese zwei Aufgaben wird auch häufig in Texterkennungsprogrammen und im Bereich der *Scene-Text-Detection* verwendet (Breuel et al., 2013; Liu et al., 2018). Allerdings ist in manchen Anwendungsfällen eine Rückkopplung vom zweiten Schritt zum ersten hilfreich, um die Erkennungsgenauigkeit zu verbessern, denn die Texterkennung kann Zusatzinformationen für das Bestimmen der Labelpositionen liefern. Eine Kombination dieser beiden Aufgaben wird z. B. in Moysset et al. (2017) und Liu et al. (2018) verwendet.

Im Folgenden werden die relevanten akademischen Publikationen in diesen beiden Teilgebieten der automatischen Erkennung von Text in Bildern ausführlich diskutiert und der eigene Beitrag zum jeweiligen Teilgebiet deutlich gemacht. Abschnitt 2.2.1 bietet einen Einblick in die Forschungsarbeiten im Teilbereich *Automatische Erkennung von Textbereichen* an. Im Abschnitt 2.2.2 werden die relevanten Forschungsarbeiten im Teilbereich *Texterkennung* besprochen.

2.2.1. Techniken zur Textlokalisierung in Bildern

Im Bereich der Textlokalisierung in Grafiken wurden sehr früh Methoden der Trennung von Text und Grafik eingesetzt, welche für jedes Objekt in einem Bild beurteilen, ob es als Textbestandteil oder Grafik eingeordnet wird. Anschließend werden die Textbestandteile zu Wörtern, Zeilen, usw. zusammengefasst. Angewendet werden dieser Techniken z.B. auf Architekturzeichnungen (Ahmed et al., 2011), technische Zeichnungen (Dori und Wenyin, 1999), Stadtpläne (Chiang und Knoblock, 2015; Cao und Tan, 2002) und Landkarten (Roy et al., 2008). Für Stadtpläne und Landkarten wurden allgemeine Algorithmen für die Trennung von Text und Grafik in Kombination mit Farbinformationen verwendet.

Zu den allgemeinen Algorithmen und Heuristiken für die Trennung von Text und Grafik gehören die Analyse der Zusammenhangskomponenten, Layoutanalyse und Texturanalyse. Für diese Ansätze werden unterschiedliche Eigenschaften der Texte in modernen und historischen Karten ausgenutzt, wie beispielsweise bestimmte Textausrichtung sowie Annahmen über die Größe der Textsymbole. Eine sehr häufig verwendete Methode ist die von Tombre et al. (2002), welche von vielen Autoren als Basis für Erweiterungen und Vergleiche gesehen wird (Roy et al., 2007; Ahmed et al., 2011). Diese Methode ist sehr generell, sie berücksichtigt aber nicht die speziellen Eigenschaften von Altkarten, wie beispielsweise vergilbtes Papier, schlechter Erhaltungszustand der Originaldokumente, verschiedene andere Symbole, die in den Größenbereich der Textsymbole fallen, sowie große Variation in Stil, Größe und Ausrichtung von Textsegmenten.

Es wurde eine große Auswahl von Texterkennungs- und Text/Grafik-Separationsalgorithmen für verschiedene Bildtypen vorgeschlagen. Ein häufiger Anwendungsfall ist die Layoutanalyse bei Texterkennungsprogrammen. Diese Systeme erwarten aber eine regelmäßige Zeilenstruktur und größere zusammenhängende Textblöcke, was aber für Texte in Karten nicht gilt. Die meisten Text/Grafik-Separationsalgorithmen basieren auf Textureigenschaften (Chen und Yuille, 2004; Gllavata et al., 2004; Kim et al., 2003; Shi et al., 2005; Shivakumara et al., 2009; Wahl et al., 1982; Yu et al., 2016) oder Eigenschaften von Zusammenhangskomponenten (Fletcher und Kasturi, 1988; Gatos et al., 2005; Kasturi et al., 1990; Tombre et al., 2002; Roy et al., 2007; Ahmed et al., 2011; Yao et al., 2012). Texturbasierte

2. Stand der Forschung

Methoden berechnen Features, z. B. lokale Helligkeitsschwankungen, DCT- oder Wavelet-Koeffizienten für jeden Punkt und verschiedene Auflösungen des Bildes. Bestimmte Charakteristiken von Text, viele Helligkeitsschwankungen, bestimmte vorherrschende Kantenrichtungen usw., spiegeln sich in diesen Features wieder und ermöglichen dadurch eine Separation von Text und Grafik. Die auf den Eigenschaften von Zusammenhangskomponenten arbeitenden Methoden filtern diese mit Hilfe von statistischen Verfahren und Heuristiken über ihre Features. Die benötigten Zusammenhangskomponenten werden mittels einer Binarisierung oder Farbsegmentierung extrahiert Roy et al. (2008); Chiang und Knoblock (2010a); Chiang et al. (2011).

Der momentan sehr aktive Bereich der *Scene Text Detection/Recognition* befasst sich mit der Erkennung von Textbereichen und der Texterkennung in Bildern und Fotos, was mit der Erkennung von Textbereichen auf einer Karte und dem Erkennen des Texts innerhalb dieser Bereiche verwandt ist.

Maximally stable extremal regions (MSER) wurde von Chen et al. (2011) für die Texterkennung eingeführt und die drei besten Methoden bei Aufgabe 4.1 (*Text Localisation of Incidental Scene Text*) der *ICDAR Robust reading Challenge* im Jahr 2014 basierten alle auf MSER (Karatzas et al., 2015). Mittlerweile wurden diese Verfahren von den Fortschritten der neuronalen Netze überholt, und es finden sich unter den Bestplatzierten nur noch Deep-Learning-Verfahren welche direkt auf den Bilddaten arbeiten.

Obwohl sich die meisten Deep-Learning-Ansätze auch in Textlokalisierung und Texterkennung aufteilen lassen, gibt es auch Ansätze, welche in einem Schritt aus dem Eingabebild einen normalisierten Text produzieren, siehe zum Beispiel Smith et al. (2016). In Smith et al. (2016) werden Schilder mit Straßennamen gelesen und diese in einem normalisierten Format, wie es als Angabe auf Karten verwendet werden würde, ausgegeben. Die meisten Systeme setzen aber im ersten Schritt auf eine Lokalisierung des Textes und setzen dann ein unabhängiges Verfahren zur OCR ein.

In He et al. (2017) werden die Deep-Learning-basierten Texterkennungsansätze eingeteilt in Ansätze die indirekte und direkte Regression verwenden. Bei der indirekten Regression (Liao et al., 2016; Liu und Jin, 2017; Liao et al., 2018) werden

2.2. Automatische Erkennung von Text in Bildern

Standard-Textboxen verwendet, welche regelmäßig über das Bild verteilt sind, und ausgehend von der ähnlichsten Box werden Offsets für die Ecken dieser Box berechnet, um sie an die Ground-Truth-Box anzupassen. Bei der direkten Regression (He et al., 2017; Zhou et al., 2017; Liu et al., 2018) werden die Eckpunkte des Textbereiches als Offset zu dem Punkt, an dem die Vorhersage erfolgt, angegeben. Eine Schwachstelle der indirekten Regression liegt in der Auswahl der Standard-Textboxen. Für jede Ground-Truth-Box muss eine ähnliche Standard-Textbox existieren, was durch das stark variierende Seitenverhältnis der Textboxen erschwert wird. Wenn zusätzlich auch rotierter Text erkannt werden soll, so benötigt dies nochmals deutlich mehr an Standard-Textboxen. Liu und Jin (2017) merken selbst an, dass die manuell erstellen Standard-Textboxen möglicherweise nicht die beste Wahl sind, und der hohe Recall ihrer Methode stark von den gewählten Standard-Textboxen abhängt.

Einen anderen Ansatz verfolgt Deng et al. (2018). Sie sagen für jedes Pixel vorher, ob es in einer Textbox liegt und außerdem für die vier benachbarten Pixel ob sie in derselben Textbox liegen. Um hier schließlich die Textboxen zu erhalten ist ein weiterer Schritt nötig, der aus den Pixel- und Verlinkungsdaten die Textboxen extrahiert. Ein weiterer interessanter Ansatz für horizontale Texte wird in Moysset et al. (2017) beschrieben. Hier wird nur die Startposition einer Zeile erkannt und dann das Texterkennungsmodul genutzt, um das Zeilenende zu erkennen. Für jede Zeile wird demnach die untere linke Ecke der Bounding-Box und deren Höhe bestimmt. Die Bounding-Box wird dann bis zum rechten Rand der Seite erweitert, und das Texterkennungsmodul führt nun eine Texterkennung bis zum Zeilenende durch. Dies wird noch durch die Verwendung eines speziellen Symbols für das Zeilenende beim Training unterstützt.

In einem Übersichtsartikel zur Trennung von Text und Grafik von Bhowmik et al. (2018) wird die Trennung von Text und Grafik noch als Methode zur Textextraktion in Karten angegeben, wobei in allen anderen Bereichen, besonders für die Erkennung in Fotos, Verfahren des maschinellen Lernens oder Deep-Learning aufgelistet werden.

Die Flexibilität der für *Scene Text Detection/Recognition* typischerweise verwendeten Deep-Learning-Ansätze bietet eine erfolgversprechende Basis für eine potentielle Übertragung dieser Ansätze auf die Altkarten. Da die hier vorgestellten Verfah-

2. *Stand der Forschung*

ren hauptsächlich für horizontale Texte ausgelegt oder trainiert sind, ist eine Erweiterung und Anpassung an die Domäne der historischen Karten notwendig. Diese Lücke wird in der vorliegenden Arbeit geschlossen. Darüber hinaus, wird die klassische regelbasierte Methode von Tombre et al. (2002), die am Anfang dieses Abschnittes diskutiert wurde, als Ausgangspunkt für den Vergleich und die Evaluierung genommen. Die Anpassung der Methode an die Domäne der Altkarten ist auch hier erforderlich und wird in der vorliegenden Arbeit unternommen. Die entsprechenden technischen Details und die Ergebnisse werden ausführlich im Abschnitt 4.1 vorgestellt.

2.2.2. **Techniken zur automatischen Texterkennung**

Als Texterkennung wird der Prozess der Erkennung und Umwandlung des in einem pixelbasierten Bild enthaltenen Textes in eine bearbeitbare zeichenbasierte Darstellung bezeichnet. Die in diesem Bereich kommerziell erhältliche Software, wie FineReader (ABBYY FineReader 14, 2018) oder OmniPage (OmniPage OCR, 2018), ist auf die Erkennung von Fließtext, wie er z.B. in Büchern vorkommt, ausgelegt.

Die ersten Texterkennungssysteme arbeiteten bei der Erkennung zeichenbasiert. Das zu erkennende Dokument wurde zuerst einer Layoutanalyse unterzogen, welche die im Dokument vorhandenen Textblöcke identifiziert. Diese Textblöcke wurden wiederum in einzelne Zeilen, Wörter und schließlich in einzelne Zeichen zerlegt. Diese Zerlegung muss nicht eindeutig sein, d. h. es können auch mehrere Hypothesen für die möglichen Zerlegungen in einem Schritt aufgestellt werden und in einem späteren Schritt z.B. anhand eines Wörterbuchs die Entscheidung für eine der Hypothesen getroffen werden. Schließlich wurde die Erkennung für die einzelnen Zeichen durchgeführt, indem sie mit bekannten Prototypen für alle bekannten Zeichen verglichen wurden. Auch frühere Versionen der OCR-Software Tesseract arbeiten nach diesem Schema. Tesseract wurde von 1984 bis 1994 von HP entwickelt, 2005 als Open-Source-Software freigegeben und wird momentan von Google weiterentwickelt. (Smith, 2007; Tesseract GitHub Repository, 2018)

Durch Erfolge in der Deep-Learning-Technik kann die Texterkennung heutzutage

2.2. Automatische Erkennung von Text in Bildern

auch in vielen Gebieten mit hoher Genauigkeit eingesetzt werden. Insbesondere die Einführung der *Connectionist Temporal Classification* (CTC) Ausgabe- und Kostenfunktion von Graves et al. (2006) in Kombination mit rückgekoppelten neuronalen Netzen (RNN) ermöglicht dies. Die CTC Ausgabe- und Kostenfunktion erleichtert das Training, indem unsegmentierte Sequenzen direkt als Eingabe verwendet werden können und die Ausgabesequenz produzieren. Für die Texterkennung kann auf diese Weise eine komplette Zeile als Eingabe verwendet werden, und es muss für das Training nur der Ausgabertext bekannt sein, nicht aber die Zuordnung zwischen den Zeichen in der Ausgabe und den Positionen im Eingabebild. Dies erleichtert einerseits das Erfassen von Trainingsdaten, da die Zuordnung zwischen Ein- und Ausgabe nicht benötigt wird, andererseits macht es dies überhaupt erst möglich, wenn es keine eindeutige Zuordnung zwischen Ein- und Ausgabe gibt. Die Handschriftenerkennung ist ein Fall, in dem eine eindeutige Trennung in einzelne Buchstaben nicht möglich ist, aber sich ein Training mit CTC einfach realisieren lässt (Graves et al., 2009).

Der Einsatz zur Erkennung von gedrucktem Text in Ocropy wird von Breuel et al. (2013) beschrieben und wird auch von Tesseract ab Version 4.0 verwendet (Tesseract 4.0 Training, 2018). Auch bei der Transkription von Texten aus mit Kameras aufgenommenen Fotos, Englisch „scene text recognition“, wurden in den letzten Jahren beeindruckende Fortschritte durch die Verwendung von RNNs in Kombination mit einer CTC-Ausgabeschicht erzielt. Beispielsweise wird dies in Su und Lu (2014) und Shi et al. (2016) eingesetzt.

Obwohl das vereinfachte Training durch die CTC Ausgabe- und Kostenfunktion es möglich macht, ohne großen Aufwand Trainingsdaten zu generieren, werden dennoch große Mengen an Trainingsdaten benötigt. So wird Tesseract beispielsweise pro Sprache mit ca. 500.000 Zeilen trainiert, und die Trainingszeit liegt im Bereich von drei bis fünf Tagen (Smith, 2016). Für Ocropy konnten jedoch für einen anderen Anwendungsfall gute Ergebnisse mit 1.000 bis 5.000 Trainingszeilen erzielt werden, siehe Springmann (2015). Im Transcribus How-To (2018) wird angegeben, dass ca. 5.000 Wörter bei gedrucktem und ca. 15.000 bei handschriftlichem Text für das Training benötigt werden. Da die Karten einige hundert bis einige tausend Ortslabels enthalten, sind in einer Karte damit nicht genügend Textlabels vorhanden um die hier genannten Mengen an Trainingsdaten zu erreichen. Aus diesem Grund ist es ein Ziel dieser Arbeit, die Menge an Trainingsdaten, die für

2. *Stand der Forschung*

die Texterkennung benötigt wird, drastisch zu reduzieren.

Hierzu werden Anpassungen der Architektur des Deep-Learning-Modells untersucht, sowie Methoden zur effizienten Verwendung der Trainingsdaten. Diese Ergebnisse werden mit Ocropy verglichen, für welches aber auch Anpassungen erforderlich sind, um ein Training mit den Daten aus den Altkarten erfolgreich durchführen zu können. Die entsprechenden Details und Ergebnisse werden im Abschnitt 4.2 vorgestellt.

2.3. Zusammenführen ähnlicher Karten

Das Zusammenführen von Karten um diese überlagern zu können wird von den meisten Geographischen Informationssystemen (GIS) in ähnlicher Weise unterstützt. Die Open-Source GIS-Software QGIS enthält hierzu ein Plugin um Rasterlayer (Rasterbilder) zu Georeferenzieren oder auch in ein neues Koordinatensystem zu transformieren. Dies wird erreicht indem Bezugspunkte (Ground Control Points (GCP)) im Rasterlayer markiert werden und diesen Geokoordinaten zugewiesen werden. Das Setzen der entsprechenden Geokoordinaten geschieht manuell, entweder durch direkte Eingabe der Koordinaten oder durch die Auswahl der entsprechenden Position in einem georeferenzierten Layer. Nachdem genügend, abhängig von der Transformationsmethode und benötigter Genauigkeit, GCPs gesetzt wurden kann eine Transformation berechnet werden die zwischen Raster- und Geokoordinaten abbildet. Je nach Grad der in den Daten enthaltenen Verzerrungen können verschiedene Typen von Transformationen gewählt werden. Wenn nur die Skalierung und Rotation angepasst werden muss, können einfachere Transformationen verwendet werden als wenn auch lokale Deformationen ausgeglichen werden müssen. (QGIS Georeferenzier Plugin, 2018)

MapAnalyst (Jenny und Hurni, 2011) ist ein Programm zur Analyse von Altkarten, welches auch die Verzerrungen in diesen Karten visualisieren kann. Wie auch in GIS-Programmen ist hier ein manuelles definieren der Kontrollpunkte nötig und Ausrichtung, Maßstab und die Verzerrung der Karte zu berechnen.

Die Georeferenzierung in GIS Systemen ist nicht automatisiert und beruht auf der

2.3. Zusammenführen ähnlicher Karten

manuellen Auswahl von Bezugspunkten. In Chen et al. (2004) und Doucette et al. (2007) werden zwei Ansätze beschrieben die ohne manuelle Zuordnung von Bezugspunkten arbeiten. Beide Arbeiten basieren darauf aus Karten das Straßennetzwerk und Kreuzungen darin zu extrahieren und dessen Entsprechung in Vektorkarten zu identifizieren. Hierzu wird in Chen et al. (2004) eine Ausrichtung beider Karten nach Norden angenommen und somit nur die Bestimmung einer Skalierung und Verschiebung zwischen den Karten benötigt. Dies lässt sich berechnen wenn zu zwei Punkten die jeweiligen Entsprechungen der anderen Karte bekannt sind. Diese Entsprechungen werden gefunden indem alle möglichen Kombinationen getestet werden bis die Anzahl der Punkte welche sich durch die gefundene Transformation aufeinander abbilden lassen einen festgelegten Schwellwert überschreitet. In Doucette et al. (2007) werden komplexere Features wie Winkel, Längen, Flächen oder Liniensegmente für die Registrierung verwendet. Zur Registrierung wurden die zwei Methoden ASA (algebraic structural algorithm) (Kovalerchuk und Chase, 2005) und STLF (similarity transformation of local features), eine Anpassung von GIPSC (Seedahmed und Martucci, 2002) auf Liniensegmente, verwendet.

Nach einem ähnlichen Prinzip arbeiten Methoden zur Bildregistrierung. Es werden markante Punkte in den Bildern bestimmt und für jeden dieser Punkte eine Beschreibung der lokalen Umgebung berechnet. Beides sollte möglichst invariant gegenüber Änderungen die zwischen den Bildern auftreten können sein, wie z.B. Koordinatentransformationen die durch verschiedene Blickwinkel entstehen können, Beleuchtungsänderungen oder Rauschen. Bekannte Algorithmen aus diesem Bereich sind SIFT (Lowe, 1999) und SURF (Bay et al., 2008). Beide Algorithmen detektieren markante Punkte und berechnen für diese rotations- und skalierungs-invariante Merkmalsvektoren die auch zu einem gewissen Grad stabil gegenüber kleinen Verzerrungen und Rauschen sind. Die berechneten Merkmalsvektoren zweier Bilder können dann genutzt werden um automatisch übereinstimmende Kontrollpunkte in den Bildern zu finden. Diese, mit Rauschen behafteten, Kontrollpunkte werden dann gefiltert und verwendet um eine Transformation zwischen den Bildern zu berechnen. Arganda-Carreras et al. (2006) beschreibt ein solches Verfahren um elastische Registrierung durchzuführen. Typische Anwendungsgebiete sind in der Fotografie, Satellitenfernerkundung, Medizin, geographischen Informationssystemen und dem maschinellen Sehen zu finden (Zitova und Flusser, 2003). Fiji (Schindelin et al., 2012), eine Open-Source-Plattform für die Bildanalyse, mit einem Schwerpunkt auf dem biologischen und medizinischen Bereich, enthält

2. *Stand der Forschung*

mehrere Bildregistrierungsalgorithmen.

Die hier vorgestellten Verfahren die auf einer manuellen Auswahl der Kontrollpunkte und ihrer Korrespondenzen in den Bildern beruhen, scheiden von vornherein aus da sie zu viel manuellen Aufwand erfordern wenn ein Kartenscan mit einer vorhanden Datenbank an Kartenscans abgeglichen werden soll. Die vorgestellten Verfahren zur Bildregistrierung können übereinstimmende Kontrollpunkte in zwei Bildern automatisch bestimmen, benötigen hierzu aber eine visuelle Übereinstimmung zwischen den Bildern. Für Karten kann es aber vorkommen, dass diese die gleichen Informationen enthalten aber in verschiedenen Stilen gezeichnet sind, diese Karten lassen sich nicht mit den Techniken der automatischen Bildregistrierung abgleichen. Sehr wohl funktioniert die automatische Bildregistrierung aber wenn es sich um zwei Scans derselben Karte oder zwei Scans von verschiedenen abdrucken derselben Druckplatte handelt. Von den zwei verbleibenden Verfahren Chen et al. (2004) und Doucette et al. (2007) setzt Doucette et al. (2007) komplexere Merkmale ein, welche auf den Beziehungen mehrere Kontrollpunkte eines Bildes aufbauen. Chen et al. (2004) basiert als einziges Verfahren alleine auf extrahierten Punkten, konkret Kreuzungen, ist aber eingeschränkt auf Verschiebungen und Skalierungen und nutzt aber einen Brute-Force-Ansatz um alle möglichen Korrespondenzen zwischen Kontrollpunkten zu testen. Eine Erweiterung der zulässigen Transformationen würde den Aufwand hier nochmals deutlich erhöhen, da mehr übereinstimmende Kontrollpunkte bestimmt werden müssen.

In Abschnitt 6 wird ein Verfahren vorgestellt welches wie Chen et al. (2004) auf Punktfeatures der Karten, hier Positionen von Orten, beruht und ausgehend von annotierten Orten iterativ neue übereinstimmende Kontrollpunkte mit anderen Karten bestimmt. Über die annotierten Orte können die ersten Kontrollpunkte mit anderen Karten aus einer Datenbank verknüpft werden und der weitere iterative Abgleich gestartet werden, somit ist dieses Verfahren auch geeignet um Karten mit einer Datenbank abzugleichen.

2.4. Georeferenzierungs- und Annotationstools für Karten

Geographische Informationssysteme sind Systeme in denen räumliche Daten, verwaltet, bearbeitet, präsentiert und analysiert werden können. Diese bieten typischerweise auch die Möglichkeit Rasterdaten zu digitalisieren, hierzu werden, wie schon in Abschnitt 2.3 beschrieben, Bezugspunkte definiert und mithilfe dieser eine Transformation der Rasterkoordinaten in ein geodätisches Referenzsystem ermöglicht. Zusätzlich können mit einem GIS-System Daten der Rasterkarte erfasst und vektorisiert werden. Hierzu stehen die Merkmalsklassen Punkte, Linien, Polygone und Annotationen zur Verfügung. So können dann beispielsweise Linien genutzt werden um Straßen und Flüsse zu erfassen, wobei die Erfassung in verschiedenen Ebenen geschieht um die Bedeutung der Linien unterscheiden zu können. Dies ermöglicht eine manuelle Erfassung aller gewünschten Daten einer Rasterkarte.

Zur Erfassung spezieller Daten existieren auch Ansätze um diese automatisch oder semiautomatisch zu erfassen, wie z.B. Strabo (Chiang und Knoblock, 2010b) zum Erfassen des Straßennetzwerks aus Stadtplänen oder auch die Vielzahl der Ansätze um aus Satelliten- und Luftaufnahmen Straßendaten zu extrahieren (Wang et al., 2016).

Auch zur Annotation und Georeferenzierung von Altkarten existieren schon Tools. Die Software-Unterstützung besteht meistens daraus, dass der Benutzer einen Punkt in der Altkarte und einen Referenzpunkt in der modernen Karte angibt oder eine Liste mit Koordinaten und deren Namen zum Durchsuchen angeboten wird. Diese Ansätze finden Verwendung in MapAnalyst (Jenny und Hurni, 2011), Georeferencer (Fleet et al., 2012), dem YUMA Map Annotation Tool (Simon et al., 2011) und Recogito (Simon et al., 2017). Jeder Automatisierungsgrad, der über die manuelle Annotation hinausgeht, trägt zum Stand der Forschung in diesem Bereich bei.

Die Georeferenzierung für moderne und historische Karten wird nicht als Problem angesehen, so wird in Dhar und Chanda (2006) angegeben, dass der Maßstab sowie der minimale und maximale Längen- und Breitengrad als Benutzereingabe erwartet wird und damit alle Pixelkoordinaten in Geokoordinaten umgerechnet

2. *Stand der Forschung*

werden können. Durch die Verwendung von Kartenprojektionen, die eine Abbildung zwischen den 2D-Koordinaten auf einer ebenen Karte und den Geokoordinaten herstellt, ist eine eindeutige Umrechnung möglich. Es genügt ein paar Punkte, z. B. die Eckpunkte, in der zu annotierenden Karte zu referenzieren um für alle anderen Punkte die Geokoordinaten berechnen zu können. So kann man mit einer kleinen Unschärfe die Koordinaten aus einer Karte in die andere umrechnen (Spektrum der Wissenschaft, 2000a). Diese Projektionsgenauigkeit ist für die Altkarten nicht gegeben.

3. Erkennung von Ortssignaturen

In diesem Kapitel wird die automatisierte Erkennung der Ortssymbole behandelt. Zunächst wird die Problematik des Altkartenmaterials als Daten für die automatische Symbol-Extraktion an Beispielen aus mehreren Altkarten ausführlich erklärt (Abschnitt 3.1). Weiter werden zwei Methoden analysiert, um die erste Forschungsfrage zu beantworten:

- ein auf dem Template-Matching (Lewis, 1995) basierender Algorithmus,
- ein auf einem Convolutional-Neural-Network (CNN) basierender Ansatz.

Im Abschnitt 3.2 wird das Template-Matching-Verfahren vorgestellt, gemeinsam mit den vorgenommenen Anpassungen, um es besser an die Bedürfnisse dieser schwierigen Art von Daten anzupassen und die Auswahl eines Schwellwerts für die Treffer zu automatisieren. Die auf einem Deep-Learning-Ansatz basierende Erkennung von Ortssignaturen wird in Abschnitt 3.3 beschrieben.

Beide Algorithmen benötigen für ihr Training Annotationen für einen kleinen Teil der Karte. Für beide Algorithmen werden die manuellen Annotationen in ein Trainings- und Validation-Set aufgeteilt. Es können die gleichen manuellen Annotationen für beide Verfahren verwendet werden, was sowohl den Vergleich der Ansätze als auch den Wechsel zwischen ihnen erleichtert. Im darauffolgenden Abschnitt 3.4 werden Evaluierungsmethodik und Ergebnisse der beiden Verfahren vorgestellt und diskutiert.

Teile dieses Kapitels wurden im Konferenzartikel (Höhn et al., 2013) auf der Joint Conference on Digital Libraries 2013 publiziert.

3.1. Altkarten als Problemfeld für die automatische Extraktion von Kartensymbolen

Die in den Altkarten als Ortssignaturen verwendeten Symbole weisen eine hohe Varianz sowohl zwischen den Karten als auch innerhalb einer Karte auf. In Altkarten wurden im Gegensatz zu modernen Karten häufig individuelle Signaturen verwendet, um verschiedene Arten der Ortschaften anzuzeigen. Manche Karten besitzen bis zu 40 verschiedene Ortssymbole, darunter alleine sechs unterschiedliche Symbole für verschiedene Arten von Mühlen, wie beispielsweise Papier-, Mehl-, oder Windmühle. Teilweise sind innerhalb bestimmter Ortskategorien (Residenz, Amtssitz, ...) die jeweils verwendeten Symbole Unikate, da sie schematische Darstellungen des Ortes sind.

Darüber hinaus waren vor dem 19. Jhdt. die Signaturen nicht standardisiert, daher können die in verschiedenen Karten für die gleichen Ortstypen verwendeten Signaturen sich unterscheiden. Besonders die Symbole für größere Ortschaften unterscheiden sich sehr in Größe und Komposition. Diese Symbole sind in Altkarten meist schematische Darstellungen von Ortschaften. Für bedeutende und seltene Ortstypen werden häufig auch schematische Darstellungen des konkreten Ortes verwendet. Beispiele hierfür sind in den Abbildungen 3.1, 3.2 und 3.3 zu sehen. Der Übergang zu den heute typischen Symbolen wie Kreis, Quadrat und Dreieck fand im 19. Jhdt. statt (Spektrum der Wissenschaft, 2000b).

In den Abbildungen 3.1a und 3.2a sind die Legenden zweier Karten dargestellt. Die Legende in Abbildung 3.1a besteht aus Zeichnungen, welche die Art des Ortes darstellen. So ist z. B. für ein Schloss der Turm eines Schlosses abgebildet, für ein Dorf wenige Gebäude, und mehrere Gebäude für eine Stadt. In dem zugehörigen Ausschnitt in Abbildung 3.1b ist allerdings zu erkennen, dass diese Symbole nicht wie erwartet auch in der Karte verwendet werden, sondern hierfür individuelle schematische Darstellungen der Ortschaften als Auf- oder Grundriss kreiert wurden. Für Würzburg beispielsweise ist eine Stadt auf einer Flussseite über eine Brücke verbunden mit einer Festung auf einem Hügel auf der anderen Flussseite abgebildet, was die Lage der Stadt am Main symbolisiert.

Die zweite Legende (3.2a) enthält schematische Darstellungen für befestigte Städte

3.1. Altkarten als Problemfeld für die automatische Extraktion von Kartensymbolen

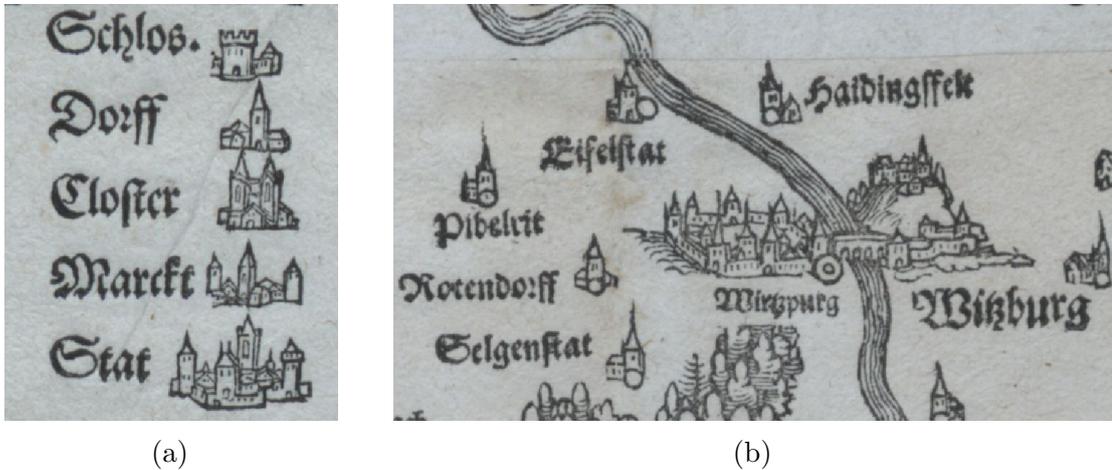


Abb. 3.1.: Legende und Ausschnitt aus Rotenhan 1533, siehe Anhang A.2.1.



Abb. 3.2.: Legende und Ausschnitt aus Goos 1626, siehe Anhang A.2.4.

und Städte, die sie auf der Karte symbolisieren sollen. Spezielle einfache Symbole sind hier vorgesehen, um Dörfer, Burgen und Klöster zu kennzeichnen. Auch hier kann man im zugehörigen Kartenausschnitt 3.2b für jede befestigte Stadt und Stadt individuelle schematische Darstellungen sehen, und nicht wie aus heutiger Sicht erwartet, überall dieselbe schematische Darstellung. Allerdings wurden für Dörfer, Burgen und Klöster die Symbole aus der Legende und keine individuellen Darstellungen verwendet.

Ein Kartenausschnitt einer weiteren Karte ist in Abbildung 3.3 zu sehen. Diese Karte enthält keine Legende. Die größeren Orte dieser Karte sind auch wieder durch individuelle schematische Zeichnungen dargestellt, und kleinere Orte durch einfachere Symbole, wobei hier alle Orte auch rot eingefärbt sind.

3. Erkennung von Ortssignaturen



Abb. 3.3.: Ausschnitt aus Merian 1656, siehe Anhang A.2.5.

Da Altkarten von Hand erstellt wurden, zeigen komplexe Symbole oft eine hohe Varianz selbst innerhalb einer einzigen Karte. Dies zeigt Abbildung 3.4, in der die für Glasfabriken verwendeten Symbole aus einer Karte zusammengefasst werden. Das Legendensymbol wird in Abbildung 3.4a gezeigt. Die Symbole, die in der Karte tatsächlich verwendet wurden, werden in den Abbildungen 3.4b bis 3.4f dargestellt.

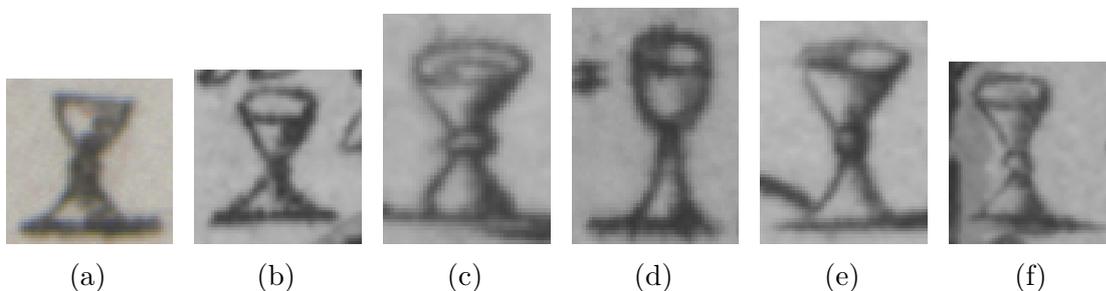


Abb. 3.4.: Variabilität der Symbole für eine Glashütte aus Fackenhofen 1791/1805, siehe Anhang A.2.7.

All diese unterschiedlichen Typen von Symbolen sollen automatisch in der Karte gefunden werden. Aufgrund der in diesem Abschnitt gezeigten hohen Variabilität sowohl zwischen verschiedenen Karten als auch innerhalb ein- und derselben Karte ist für dieses Problem keine universelle Lösung zu erwarten, die sich auf alle Sonderfälle anwenden lässt. Es sollen aber die am häufigsten auftretenden Fälle erfasst werden, um eine möglichst große Arbeitersparnis bei der Annotation zu erzielen.

3.2. Template-Matching-basierter Ansatz

Die Suche nach den in der Karte vorhandenen Ortssignaturen wird über einen Template-Matching-Algorithmus realisiert. Der Algorithmus ist ein Standardverfahren aus dem Gebiet der Bildverarbeitung. Die Neuheit der Ansatzes der vorliegenden Arbeit besteht darin, das Template-Matching-Verfahren auf die Altkarten zu übertragen, entsprechend anzupassen und die Auswahl der Konfigurationsparameter zu automatisieren. Das System erstellt hierzu automatisch Template-Varianten, die mittels der normalisierten Kreuzkorrelation mit der Karte verglichen werden. Der beste Ansatz wird anhand der Auswertung auf einem Validation-Set ausgewählt.

Bei Template-Matching wird eine Vorlage (Template) T über das Bild I bewegt und an jeder Position (x, y) über eine Abstandsfunktion die lokale Übereinstimmung bestimmt, siehe Abbildung 3.5. Dadurch ergibt sich ein Distanzbild, in welchem die Minima oder Maxima, je nach verwendeter Abstandsfunktion, die Positionen mit den besten Übereinstimmungen anzeigen. Burger und Burge (2009) verwenden die Summe der absoluten Differenzen (SAD), die maximale Differenz (max) und die Summe der quadrierten Abstände (SSD) als grundlegende Abstandsdefinitionen. Zusätzlich wird noch der Korrelationskoeffizient vorgestellt und dessen Zusammenhang mit der Summe der quadrierten Abstände aufgezeigt.

Der Korrelationskoeffizient aus Burger und Burge (2009) wird in anderen Arbeiten meist als normalisierte Kreuzkorrelation bezeichnet, z.B. in Lewis (1995) und Briechle und Hanebeck (2001), wobei Burger und Burge (2009) die Bezeichnung normalisierte Kreuzkorrelation für eine abweichende Formel verwenden. In dieser Arbeit wird der Korrelationskoeffizient aus Burger und Burge (2009) wie in Lewis (1995) und Briechle und Hanebeck (2001) als normalisierte Kreuzkorrelation (NCC) bezeichnet. Wegen ihrer Robustheit gegenüber Änderungen der Helligkeit oder des Kontrasts wird hier die normalisierte Kreuzkorrelation, siehe Gleichung 3.1, als Abstandsfunktion verwendet.

$$d_{NCC}(x, y) = \frac{\sum_{(u,v) \in M} (I_{x,y}(u, v) - \overline{I_{x,y}^M}) \cdot (T(u, v) - \overline{T^M})}{\sigma_{I_{x,y}^M} \sigma_{T^M}} \quad (3.1)$$

3. Erkennung von Ortssignaturen

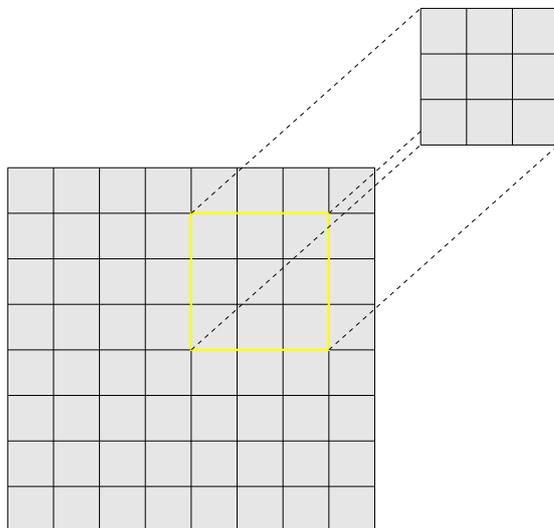


Abb. 3.5.: Templatematching mittels Sliding-Window-Verfahren. Das Template wird pixelweise über das Bild bewegt und an jeder Position über eine Abstandsfunktion die lokale Übereinstimmung bestimmt

mit

$$I_{x,y}(u, v) = I(u + x, v + y) \quad (3.2)$$

$$\overline{I}_{x,y}^M = \frac{1}{|M|} \sum_{(u,v) \in M} I_{x,y}(u, v) \quad (3.3)$$

$$\overline{T}^M = \frac{1}{|M|} \sum_{(u,v) \in M} T(u, v) \quad (3.4)$$

$$\sigma_{TM} = \sqrt{\frac{1}{|M|} \sum_{(u,v) \in M} (T(u, v) - \overline{T}^M)^2} \quad (3.5)$$

Typischerweise wird für das Template eine rechteckige Form verwendet, womit M die Menge aller Pixelkoordinaten des Templates ist. Es sind aber auch beliebige Einschränkungen von M möglich um Bereiche beim Matching zu ignorieren.

Die Erkennungsgenauigkeit wird von verschiedenen Faktoren beeinflusst. Die linearen Änderungen der Helligkeit oder des Kontrasts wirken sich auf die NCC nicht aus. Jedoch nehmen Transformationen wie z. B. Rotation und Skalierung Einfluss auf die NCC. Zusätzlich gibt es in Altkarten Symbole, welche in Teilberei-

3.2. Template-Matching-basierter Ansatz

chen übereinstimmen, siehe z. B. die Ortssignaturen in Abbildung 3.6a und 3.6b. In diesem Fall hängt die Erkennungsgenauigkeit von der Reihenfolge, in der die verschiedenen Symbole erkannt werden, ab. Wenn zuerst die Übereinstimmungen mit dem Template wie in Abbildung 3.6b ausgeführt werden und dann mit dem Template wie in Abbildung 3.6a, dann können durch das erste Template weitere Treffer an der selben Stelle blockiert werden. Da aber die komplexeren Symbole eine große Variabilität aufweisen, sind sie für ein pixelweise arbeitendes Template-matching nur schlecht geeignet. Eine Alternative ist es, nur den Ortskringel (siehe Abbildung 3.6b) als Template zu verwenden, da er auch Bestandteil jeder anderen Ortssignatur ist.

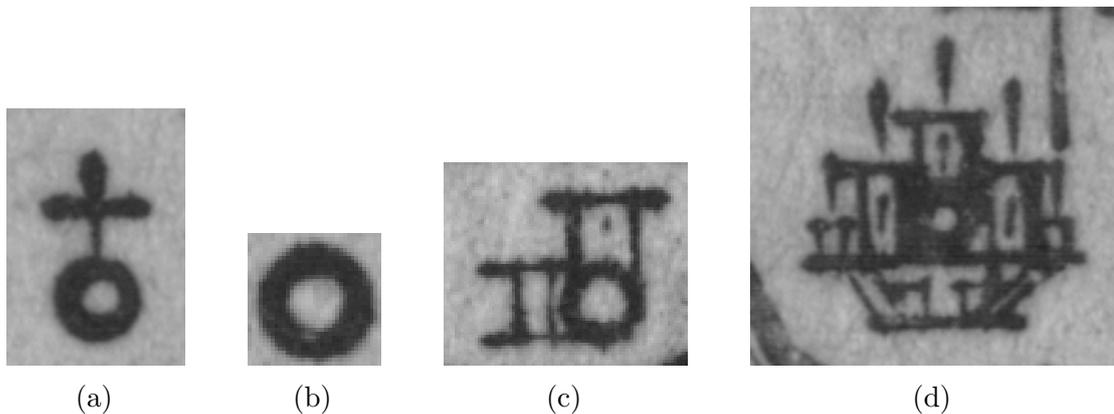


Abb. 3.6.: Ortssignaturen für Kloster, Dorf, Flecken und Stadt (links nach rechts) aus Seutter 1731, siehe Anhang A.2.6.

Durch die Verwendung von nicht rechteckigen Templates lässt sich das Template-Matching verbessern, da sich der Bereich der beachtet wird, besser einschränken lässt. Für die hier betrachteten Altkarten ist dies besonders wichtig, da sehr oft andere Elemente wie Straßen, Flüsse, Ortsnamen usw. in den rechteckigen Bereich um die Ortssignatur hineinreichen und das Template-Matching verschlechtern würden. Das Erstellen einer Maske würde aber einen großen Zusatzaufwand darstellen und kleine Änderungen an der Maske können große Auswirkungen auf die Qualität des Matchings haben. Daher müssten verschiedene Masken erstellt werden und getestet werden. Da dies zu aufwendig wäre, wurde dieser Schritt automatisiert. Es werden verschiedene Masken automatisch erstellt und die Matchingqualität in einem vom Benutzer annotierten Bereich überprüft.

3. Erkennung von Ortssignaturen

Für die Erstellung der Masken wird eine binarisierte Version des Templates verwendet, da die Symbole in den Altkarten Schwarz auf Weiß gedruckt wurden und so der bedruckte Teil des Symbols erfasst wird aber alles um das Symbol herum das Matching nicht beeinflusst. In Karten mit Symbolen mit scharfen Rändern wäre dieses Verfahren nicht praktikabel, da dann die Maske die schwarze Fläche erfasst und diese in jedem schwarzen Bereich der größer ist wiederfinden würde. In den betrachteten Altkarten tritt am Rand immer ein Übergang von Schwarz nach Weiß mit Grauabstufungen auf, so dass der Rand mit in das Template eingeht. Dieser Effekt lässt sich noch verstärken, wenn die Maske (das binarisierte Symbol) erweitert wird, d. h. jedes Pixel, das in der direkten Nachbarschaft ein schwarzes Pixel hat, wird auch schwarz gefärbt. Dadurch wird mehr von der Kante erfasst, es wird aber auch die Gefahr erhöht, dass benachbarte Elemente in den Bereich der Maske hineinreichen. Daher werden verschiedenen Masken erstellt, die verschieden oft erweitert wurden und so verschieden große Bereiche in der Umgebung des Symbols mit abdecken. Da der Bereich im Inneren des Symbols normalerweise weiß ist, wird von jeder Maske noch eine Variante erstellt, bei der die inneren Bereiche mit zur Maske hinzugenommen wurden.

Es werden sieben Masken generiert: eine Maske für das komplette Template und sechs Masken aus einer binarisierten Version des Templates $Bin(T)$. Die binarisierte Version des Templates und die ein- und zweimal erweiterte Version des Templates sowie die drei Varianten dieser Masken mit gefüllten Löchern werden als Masken verwendet. Ein Beispiel für die so erstellten Masken wird in Abbildung 3.7 gezeigt.

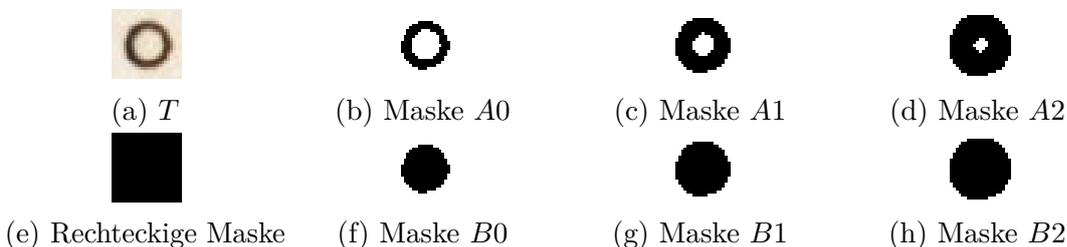


Abb. 3.7.: Abbildung 3.7a zeigt ein Template, Abbildungen 3.7b bis 3.7h die dazu automatisch erstellten Masken.

Die Selektion des besten Templates geschieht nun durch die Auswertung, mit welchem Template sich in einem Testbereich R_T der beste F1-Score erreichen lässt. Hierzu wird für jedes Template der beste Schwellwert in diesem Bereich bestimmt.

3.2. Template-Matching-basierter Ansatz

Um dies durchzuführen, werden die Annotationen im vergrößerten Bereich R_A benötigt (siehe Abbildung 3.11).

Abbildung 3.8b zeigt ein Beispiel der NCC für den Kartenausschnitt aus Abbildung 3.8a mit dem Template 3.6b. Das Korrelationsbild enthält Werte im Bereich von -1.0 bis 1.0 , mit 1.0 (Weiß) für perfekte Übereinstimmung und -1.0 (Schwarz) für einen invertierten Treffer.



(a) Kartenausschnitt aus Seutter 1731, siehe Anhang A.2.6

(b) normalisierte Kreuzkorrelation

Abb. 3.8.: Abbildung 3.8b zeigt die normalisierte Kreuz-Korrelation des Templates in Abbildung 3.6b angewandt auf Abbildung 3.8a.

Große Werte für den NCC stehen für eine gute Übereinstimmung zwischen der Vorlage und dem Bild an dieser Stelle. Dadurch lassen sich die besten Trefferpositionen für das Template mit einem Schwellenwert bestimmen. Der NCC ist aber auch in der Nähe des Matches hoch und so werden nur lokale Maxima betrachtet. Dieses Verfahren ist auch als non-maximum suppression (NMS) bekannt. Durch Abweichungen beim Druck oder Rauschen können für ein Template auch sehr nahe benachbarte lokale Maxima entstehen. Um diese Mehrfacherkennungen zu unterdrücken, werden die Treffer absteigend nach ihrem NCC sortiert, und für jeden Treffer werden alle anderen Treffer mit einem niedrigeren NCC und einem Abstand unter r_{Block} entfernt. Der Abstand r_{Block} wird standardmäßig auf die halbe Seitenlänge der kürzeren Seite des Templates gesetzt und kann vom Benutzer angepasst werden. Hierdurch entsteht eine Liste von Treffern mit dem zugehörigen NCC, welche aus den bereinigten lokalen Maxima des Korrelationsbilds bestehen.

3. Erkennung von Ortssignaturen

Alle lokalen Maxima im Korrelationsbild, die über dem Schwellwert liegen, werden als Positionen von Ortssignaturen ausgewählt. Der Schwellwert muss für jede Karte und jedes Template angepasst werden um Precision und Recall zu optimieren. Aus der Trefferliste können nun die in die Karte zu übernehmenden Ortspositionen über einen Schwellwert bestimmt werden. Dieser Schwellwert kann entweder vom Nutzer direkt vorgegeben werden, oder es kann ein Teilbereich der Karte von Hand annotiert werden, und es wird dann automatisch aus allen Template-Masken-Kombinationen der Schwellwert mit dem besten F1-Score ermittelt. Die Ermittlung des F1-Score geschieht durch den Vergleich der automatisch ermittelten Ortsposition mit Ground-Truth-Werten, welche manuell annotiert wurden.

Die Details über die Ermittlung des F1-Score und die Auswertung des hier vorgestellten modifizierten Template-Matching-Verfahrens werden im Abschnitt 3.4.2 beschrieben.

3.3. Deep-Learning-basierter Ansatz

Seit dem Sieg von AlexNet (Krizhevsky et al., 2012) in der *ImageNet Large Scale Visual Recognition Challenge 2012* sind Convolutional Networks und Deep-Learning aus dem Bereich des maschinellen Sehens nicht mehr wegzudenken (LeCun et al., 2015). AlexNet reduzierte die Top-5-Fehlerrate des Vorjahres von 26% auf 15% und auch der Zweitplatzierte, welcher kein Deep-Learning verwendete, lag noch bei der Fehlerrate des Vorjahres. Daher wurde auch hier die Ortssignaturerkennung mittels eines CNN getestet.

Für die Erkennung der Ortssignaturen wurde ein CNN mit drei Convolutional Layern (siehe Abbildung 3.10) verwendet. Zum Training wird ein Kartenausschnitt benötigt, in dem alle vorkommenden Ortssignaturen markiert sind. Aus diesem Kartenausschnitt können dann positive und negative Beispiele für das Training generiert werden. Die positiven Beispiele entsprechen den annotierten Positionen und die negativen zufälligen nicht annotierten Positionen (siehe Abbildungen 3.9).

Im hier implementierten CNN werden nur Convolutional- und Maximum-Pooling-

3.3. Deep-Learning-basierter Ansatz

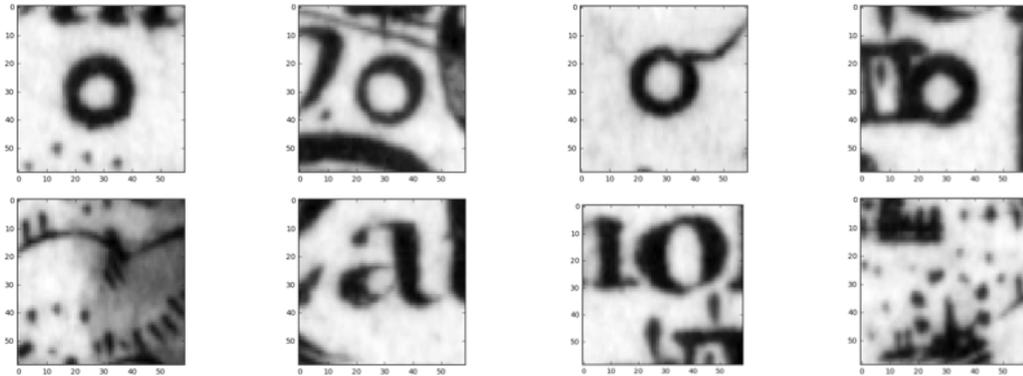


Abb. 3.9.: Positive und negative Beispiele um das CNN zu trainieren. Die obere Reihe zeigt vier positive Beispiele und die untere Reihe zeigt vier negative Beispiele. Die Trainingsdaten werden automatisch aus einem annotierten Kartenausschnitt generiert.

Layer verwendet. Da es in diesem Fall ein großes Ungleichgewicht zwischen positiven und negativen Trainingsbeispielen gibt, werden Data-Augmentation-Techniken verwendet um die Menge der positiven Beispiele künstlich zu vergrößern. Für die negativen Beispiele wird *Hard-Negative-Mining* verwendet, da viele Hintergrundbereiche leicht zu klassifizieren sind und damit nicht zum Training beitragen.

Zur Erkennung der Ortssignaturen nach dem Training wird das CNN auf ein Eingabebild angewendet und liefert für jeden Pixel einen Wert zwischen 0 und 1 zurück. Je besser das CNN die Übereinstimmung mit einer Ortssignatur einschätzt, desto höher ist der Wert. Um nun die Positionen der Ortssignaturen zu bestimmen werden wie beim Template-Matching auch nur die lokalen Maxima betrachtet, dies entspricht einer non-maximum suppression (NMS). Analog zum Template-Matching wird dann auch ein Schwellwert bestimmt, welcher den F1-Score auf dem annotierten Teilbereich optimiert.

Die Architektur des CNNs zur Erkennung von Ortssignaturen wird in Abbildung 3.10 gezeigt. Es besteht aus zwei *Convolutional Layern* (rot dargestellt), die jeweils von einem *Activation Layer* (gelb) und einem *Pooling Layer* (blau) gefolgt werden. Abschließend folgt noch ein weiterer *Convolutional Layer* und ein *Softmax Layer* (rosa). Für die *Convolutional Layer* sind jeweils die Breite (w) und Höhe (h) des Filterkerns, sowie die Schrittweite (engl.: *stride*) (s) und die Anzahl Filterkernel (engl.: *channels*) (c) im Format $w \times h/s, c$ angegeben. Die Schrittweite bewirkt

3. Erkennung von Ortssignaturen

eine Unterabtastung (engl.: *subsampling*), indem nur in jedem s -ten Schritt eine Ausgabe berechnet wird. Eine nicht lineare Aktivierungsfunktion wird auf die Ausgabe des *Convolutional Layers* angewendet, hier die *Rectified Linear Unit* (ReLU) $f(x) = \max(0, x)$ (Glorot et al., 2011) verwendet. Im *Pooling Layer* wird ein Maximum-Pooling mit einer Fenstergröße von 2×2 Pixeln und einer Schrittweite von 2 verwendet. Dies bedeutet die Eingabe wird in nicht überlappende 2×2 -Blöcke eingeteilt und für jeden Block das jeweilige Maximum als Ausgabe verwendet. Der abschließende *Softmax Layer* normalisiert die Ausgabe auf einen Bereich von 0 bis 1.

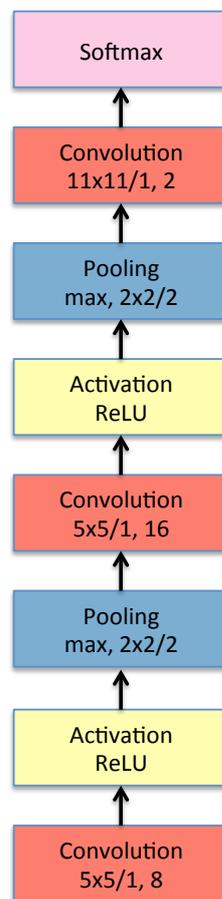


Abb. 3.10.: Aufbau des CNNs zur Erkennung von Ortssignaturen.

Für das Training wird ein Verhältnis von eins zu fünf für die positiven und negativen Trainingsbeispiele verwendet. Für die negativen Trainingsbeispiele wird auch ein *Hard-Negative-Mining* verwendet, welches die Positionen, die fälschlicherweise als Treffer klassifiziert werden, bevorzugt als negative Trainingsbeispiele verwendet.

Maximal ein Fünftel der Negativbeispiele wird durch das *Hard-Negative-Mining* bestimmt und der Rest mit zufälligen Negativbeispielen aufgefüllt. Für ein Trainingslauf werden 500 Epochen bei einer Batch-Größe von 24 verwendet, d. h. pro Epoche werden alle positiven Trainingsbeispiele einmal für das Training verwendet und es werden jeweils 4 positive mit 20 negativen Beispielen kombiniert. Diese Parameter wurden vor den Experimenten festgelegt und erlauben ein stabiles Trainieren. Bei größeren Abweichungen von diesen Parametern konnte das Training in vielen Fällen nicht konvergieren.

3.4. Experimente

Zur Evaluierung wurden in vier der Altkarten (Seutter 1731 A.2.6, Ortelius/Roteman 1570 A.2.2, Merian 1656 A.2.5, Mercator 1600 A.2.3 und zwei verschiedenen Abdrucken von Ortelius 1579 A.1.5) aus dem zur Verfügung stehenden Datensatz alle Orte markiert.

3.4.1. Experimentaufbau

Es werden sowohl die Ergebnisse des Template-Matching-Verfahrens mit dem Deep-Learning-basierten Verfahren verglichen als auch der Einfluss der Templategröße und die Effektivität der Maskengenerierung untersucht. Zum Testen der Effektivität der vorgestellten Maskengenerierung wurde ermittelt, wie sehr sich der F1-Score durch die automatisch generierten Masken gegenüber dem rechteckigen Template verbessern lässt. Dazu wurden für die Karten Templates ausgewählt und die dazugehörigen Masken erzeugt. Mit den Template-Masken-Kombinationen wurde dann das Template-Matching durchgeführt und jeweils der Schwellwert ermittelt mit dem sich der beste F1-Score ergibt. Auch wird die Auswirkung der Größe des Templates, d. h. die Größe des Randes um das Template, auf das Matching mit und ohne Masken untersucht. Nachfolgend sind die Ergebnisse aufgelistet.

3. Erkennung von Ortssignaturen

3.4.2. Evaluierungsmethodik

Für die Evaluierung der Genauigkeit der hier vorgestellten Methoden zur automatischen Identifikation und Klassifikation von Kartensymbolen werden die Metriken $precision = \frac{tp}{tp+fp}$ und $recall = \frac{tp}{tp+fn}$ verwendet. Hier steht tp für die Anzahl der *true positive*, fp für die Anzahl der *false positive* und fn für die Anzahl der *false negative*. Diese Parameter sind wie folgt definiert:

true positive: Eine Ortsposition, für die im Umkreis von r Pixeln ein Match gefunden wurde.

false positive: Ein Match, für den im Umkreis von r Pixeln keine Ortsposition definiert ist.

false negative: Eine Ortsposition, für die im Umkreis von r Pixeln kein Match gefunden wurde.

Der $F1-score = 2 \frac{precision \times recall}{precision + recall}$ ist ein Maß welches $precision$ und $recall$ mittels des harmonischen Mittels kombiniert, um durch nur einen Wert, eine leichtere Vergleichbarkeit zu ermöglichen.

Eine Toleranz von r Pixeln wird verwendet, da die exakten Pixelkoordinaten, an denen annotiert wird, beim automatischen Annotieren durch Rauschen und beim manuellen Annotieren durch Ungenauigkeiten leicht variieren können. Dies ist in Abbildung 3.11 illustriert. *True positives* sind die in grün dargestellten Ortspositionen. Diese Ortspositionen enthalten im Umkreis von r Pixeln einen Match. Die schwarz dargestellten Matches liegen im Umkreis von r Pixeln um einen Ort und gehören damit zu einem *true positive*. Die in rot dargestellten Matches enthalten im Umkreis von r Pixeln keine Ortsposition und sind damit *false positives*. Die in rot dargestellten Ortspositionen enthalten im Umkreis von r Pixeln keinen Match. Sie wurden also nicht automatisch gefunden, und sind damit *false negatives*.

Für die Auswertung der Genauigkeit ist eine Festlegung auf eine der beiden Möglichkeiten (Match oder Ortsposition) als *true positive* nötig, da sie an verschiedenen Positionen liegen können und da bei der Betrachtung eines Bereiches ein Marker innerhalb und einer außerhalb liegen kann. Die Wahl fiel hier auf die Ortspositionen

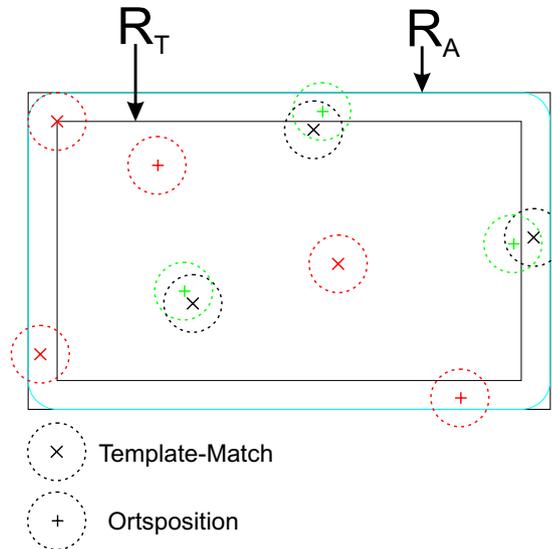


Abb. 3.11.: Auswertung der Ortserkennung. Grün dargestellte Orte wurden gefunden, rot dargestellte Orte wurden nicht gefunden und rot markierte Template-Matches entsprechen keinem Ort. Der betrachtete Testbereich ist R_T , wobei Annotationen für den Bereich R_A benötigt werden.

als Repräsentanten der *true positives*, da sie aus der Ground-Truth-Menge stammen, und mit dieser Festlegung maximal so viele *true positives* wie Ortspositionen vorkommen können. Würden die Matches als *true positives* gewählt, dann könnten beliebig viele in einem Umkreis von r Pixeln um einen Ortsposition liegen. Diese alle wären dann ein *true positive*.

Bei der Wahl der Ortspositionen als Repräsentanten kann es aber auch zu Ambiguitäten kommen. Wenn zwei Ortspositionen weniger als $2r$ Pixel Abstand haben, dann überlappen sich die Bereiche, in denen Matches akzeptiert werden. Damit kann ein Match zu mehreren gefundenen Ortspositionen führen. Solche Mehrdeutigkeiten sind nicht erwünscht. Dieses Problem lässt sich allerdings bereits an den Ground-Truth-Daten erkennen. In den meisten Fällen lässt sich diese Art von Ambiguität ausschließen, da r groß genug gewählt wird, um Abweichungen bei der manuellen Annotation zuzulassen, aber nicht größer, als nötig. Allerdings muss man für solche Fälle bedenken, dass wenn zwei Orte näher zusammenliegen als r , ist selbst eine manuelle Annotation nicht ohne Mehrdeutigkeiten gewährleistet und der subjektiven Entscheidung des menschlichen Annotators überlassen ist.

3. Erkennung von Ortssignaturen

Um für einen Testbereich R_T die automatische Annotation auszuwerten, müssen wegen der Toleranz von r Pixeln die Ergebnisse der Annotation und die Ground-Truth für einen Bereich der R_T um mindestens die Toleranzzone erweitert werden. Dies in Abbildung 3.11 als ein an allen Seiten um r Pixel vergrößerte Rechteck R_A in Blau dargestellt.

3.4.3. Auswirkung der Templategröße

Die Masken für Templates wurden eingeführt, da dicht um die Ortssignaturen andere Objekte, wie Straßen oder Ortsnamen, vorkommen und diese nicht beim Template-Matching beachtet werden sollten und auch die Effekte wie viel Rand der Benutzer um ein Template auswählt, minimiert werden sollten. Im ersten Schritt erfolgt eine Binarisierung um nur den bedruckten Teil der Ortssignatur zu erhalten. Dies führt aber dazu, dass mit dieser Maske auch das Matching in dunklen Flächen, die das Template nicht erhalten, verbessert wird. Da die hellen Bereiche, die das Matching hier berechtigterweise verschlechtern würden, durch die Maske ausgeblendet werden. Um dem entgegenzuwirken wird die Maske erweitert, d. h. für jedes schwarze Maskenpixel werden die umliegenden acht Nachbarpixel auch schwarz gefärbt. Dadurch schließt die Maske mehr helle Pixel ein und verschlechtert das Matching in dunklen Flächen, erhöht aber auch wieder die Wahrscheinlichkeit, dass andere Objekte in den Bereich der Maske gelangen. Es werden hier die ein- und zweifache Erweiterung des binarisierten Templates verwendet und die Masken als A_0 , A_1 und A_2 bezeichnet, wobei die Zahl in der Bezeichnung angibt wie oft das binarisierte Template erweitert wurde. Zusätzlich werden noch Masken verwendet, bei denen die im Template enthaltenen Löcher gefüllt wurden. Dies erhöht die zum Matching zur Verfügung stehende Fläche bei geringem Risiko durch Störeinflüsse anderer Objekte. Diese Masken werden aus den A_n -Masken durch Füllen der Löcher generiert und dann als B_n bezeichnet, mit $n \in \{0, 1, 2\}$, siehe Abbildung 3.7.

Der Matching-Score hängt nur von den Werten der Pixel im schwarzen Bereich der Maske ab und daher bleiben die Matching-Scores gleich, wenn das Template vergrößert wird und die Maske entsprechend um weiße Pixel erweitert wird. Trotz der gleichbleibenden Matching-Scores bei vergrößerten Templates und um entspre-

chende weiße Pixel erweiterte Masken gibt es hier zwei Einflussfaktoren, die die ermittelten Matches beeinflussen. Einmal müssen die Masken nicht gleich bleiben, da der Schwellwert zur Binarisierung aus dem selektierten Template berechnet wird und bei verschiedenen großen Rändern die Eingabe für den Binarisierungsalgorithmus dadurch verschieden ist. Der zweite Faktor ist die Unterdrückung überlappender Matches, welche die Größe des Templates verwendet und somit eine direkte Abhängigkeit vom Template hat. Im Folgenden werden die Auswirkungen dieser Faktoren auf das Template-Matching untersucht.

Hier wird die Auswirkung der Größe des Randes um das Template untersucht. Dazu wurde eine Ortssignatur mit möglichst großem Abstand zum nächsten Objekt in der Karte gesucht. Für dieses Experiment wurde ein Ortskringel aus der Mercator-1600-Karte A.2.3 verwendet, welcher einen Durchmesser von ca. 20 Pixeln hat und einen Rand von 25 Pixeln auf allen Seiten (siehe Abbildung 3.12). Das Matching wurde dann für Ausschnitte des Templates mit verschieden großem Rand durchgeführt. In Abbildung 3.13 wird für die verschiedenen Masken der jeweils erreichte F1-Score in Abhängigkeit von der Templategröße dargestellt. Bei der Verwendung des kompletten rechteckigen Templates („T“ in Abbildung 3.13) hängt die Genauigkeit stark von der Templategröße ab, wohingegen das modifizierte Verfahren eine deutlich geringere Abhängigkeit von der Templategröße zeigt.



Abb. 3.12.: Template mit breitem Rand aus der Mercator 1600 Karte A.2.3.

3.4.4. Ergebnisse

Die Ergebniswerte für *Precision*, *Recall*, und F1-Score für die untersuchten Karten sind in Tabelle 3.1 aufgelistet.

3. Erkennung von Ortssignaturen

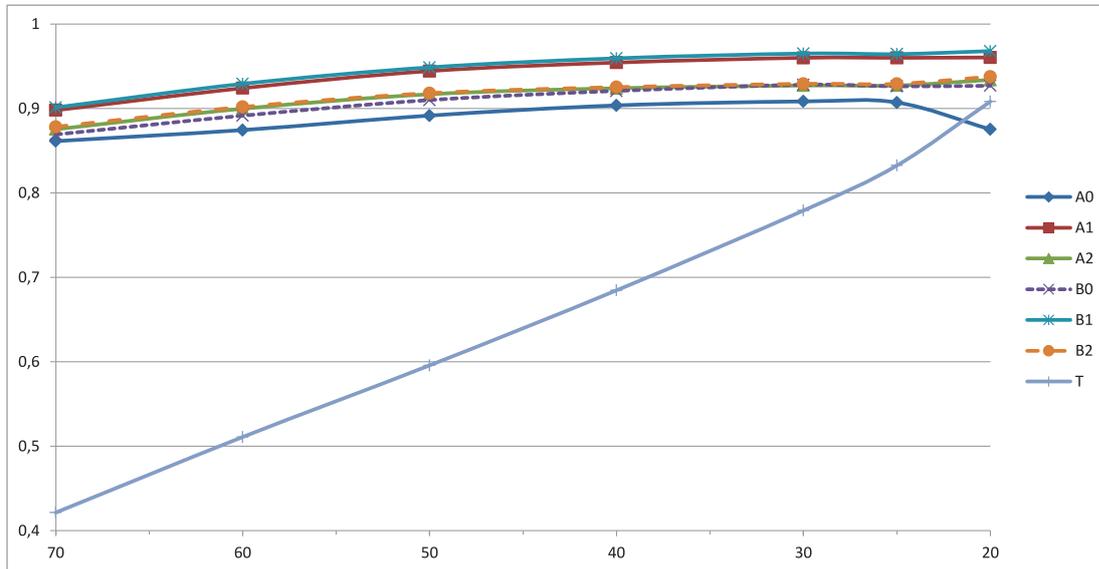


Abb. 3.13.: F1-Score in Abhängigkeit von der Templategröße. Auswertung für das Template mit verschieden großem Rand auf der kompletten Mercator 1600 Karte A.2.3.

	Template-Matching			Mod. Template-Matching			CNN		
	P	R	F1	P	R	F1	P	R	F1
Seutter 1731 A.2.6	0,846	0,772	0,807	0,982	0,877	0,927	0,932	0,884	0,907
Ort./Rot. 1570 A.2.2	0,104	0,051	0,069	0,753	0,526	0,620	0,570	0,665	0,614
Merian 1656 A.2.5	0,888	0,450	0,598	0,890	0,472	0,617	0,747	0,738	0,743
Mercator 1600 A.2.3	0,934	0,588	0,722	0,939	0,964	0,952	0,951	0,868	0,908
Ortelius 1592 Z50 A.1.5	0,410	0,201	0,269	0,725	0,816	0,768	0,892	0,901	0,897
Ortelius 1612 Z60 A.1.5	0,626	0,668	0,646	0,897	0,642	0,748	0,906	0,833	0,868

Tab. 3.1.: Precision (P), Recall (R) und F1-Score (F1) für das Template-Matching mit rechteckigen Templates, das modifizierte Template-Matching und die Ortserkennung mit CNNs.

Ergebnisse für die Seutter 1731 Karte

Folgende Ergebnisse wurden für die Seutter 1731 A.2.6 Karte erzielt.

In Tabelle 3.2 werden die Ergebnisse des Template-Matchings auf einem Ausschnitt der Seutter 1731 Karte A.2.6 gezeigt. Es sind jeweils für jede der sechs automatisch generierten Masken und für das volle Template der beste erreichbare F1-Score angegeben und der Schwellwert, mit dem der beste F1-Score erreicht wird. Der betrachtete Ausschnitt ist 1366×1006 Pixel (1,37 MP) groß und enthält 52

Tab. 3.2.: Template-Matching Ortskringel Seuter 1731 auf Ausschnitt

Maske	Bester F1-Score	Schwellwert
A0	0,91954023	0,6
A1	0,906976744	0,68
A2	0,888888889	0,64
B0	0,967032967	0,68
B1	0,909090909	0,65
B2	0,896551724	0,66
T	0,870588235	0,71

Tab. 3.3.: Template-Matching Ortskringel Seuter 1731

Maske	Bester F1-Score	Schwellwert
A0	0,901571547	0,54
A1	0,902432283	0,61
A2	0,891823546	0,67
B0	0,927744071	0,67
B1	0,900868104	0,65
B2	0,890618367	0,68
T	0,83729485	0,71

Ortssignaturen. Mit dem Template lässt sich der beste F1-Score durch Verwendung der B0 Maske mit einem Schwellwert von 0,68 erreichen. Diese Kombination aus Maske und Schwellwert wird nachfolgend auch verwendet um die Ortssignaturen in der gesamten Karte zu extrahieren.

Der annotierte Ausschnitt umfasst 0,81% der Kartenfläche von 14283×11878 Pixel (169,65 MP) und 1,28% der 3734 Ortssignaturen. Schon mit dieser kleinen Menge an annotierten Orten lässt sich eine gute Kombination aus Maske und Schwellwert ermitteln, welche auf dem Gesamtbild einen F1-Score von 0,926470588 erreicht, was besser ist als alles, was sich mit den restlichen Masken erreichen lässt und nur 0,1273482 Prozentpunkte vom mit dieser Methode möglichen Optimum entfernt ist. In absoluten Zahlen bedeutet dies, es werden auf der Gesamtkarte 3399 Orte markiert, wovon 3339 richtig sind und 60 falsch. Dies entspricht bei 3809 Orten einer Precision von 0,982347749 und einem Recall von 0,876608034. Das optimale Ergebnis liegt bei 3443 markierten Orten, von denen 3364 richtig und 79 falsch sind.

3. Erkennung von Ortssignaturen

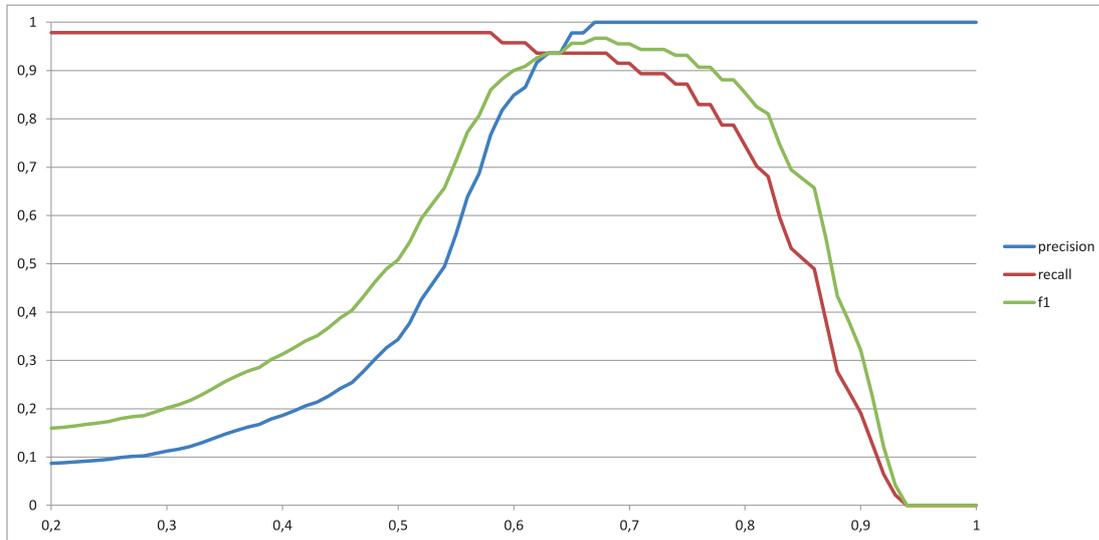


Abb. 3.14.: Auswertung des Templatematching mit Maske B0 auf einem Ausschnitt der Seutter 1731 A.2.6 Karte.

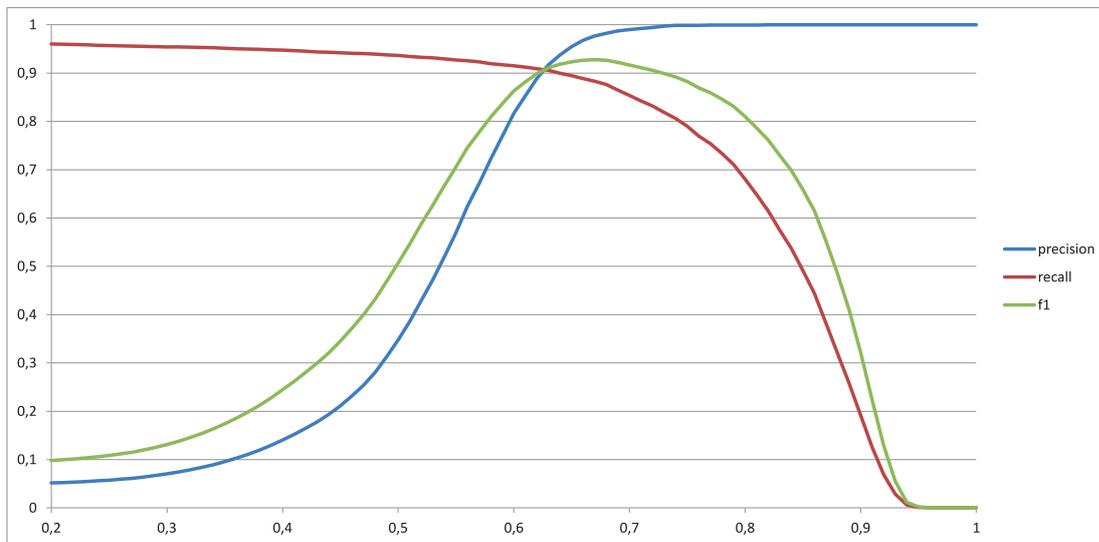


Abb. 3.15.: Auswertung des Templatematching für das Template mit Maske B0 auf der kompletten Seutter 1731 A.2.6 Karte.

Abbildung 3.16 zeigt Precision und Recall für die Erkennung von Ortssignaturen mittels des in Abschnitt 3.3 vorgestellten Verfahrens auf der Seutter 1731 A.2.6 Karte. Bei der Verwendung des CNNs wurde der Kartenrand und die Verzierungen von der Erkennung ausgeschlossen, da diese viele falsche Treffer produzieren. Ein Grund für die vielen falschen Treffer in diesen Bereichen ist, dass sie nicht in den

Trainingsdaten vorkommen und das Netzwerk ihre Klassifizierung damit nicht lernen kann.

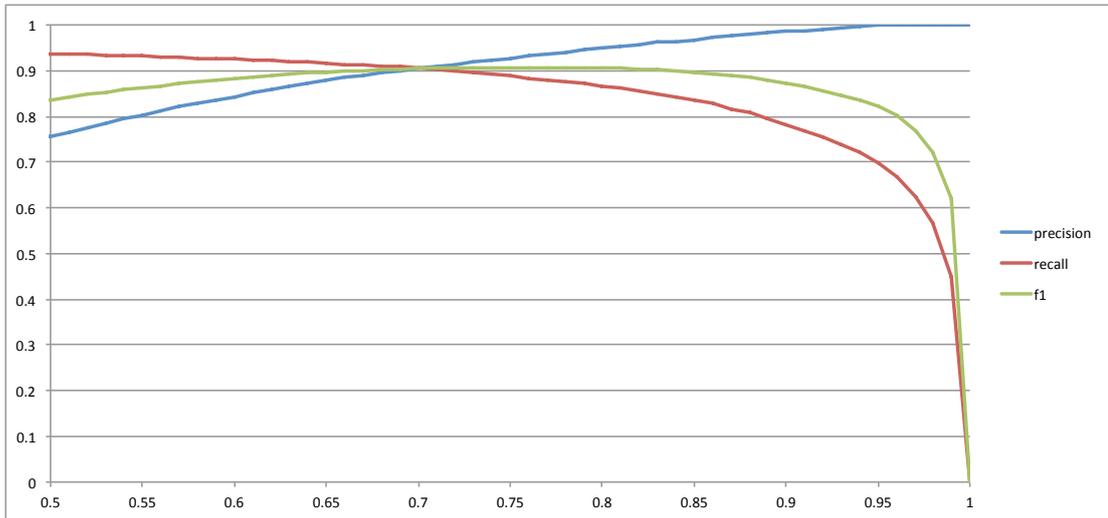


Abb. 3.16.: Auswertung der Erkennung von Ortssignaturen mittels CNN auf der Seutter 1731 A.2.6 Karte.

In Abbildung 3.17 wird ein Vergleich der Precision-Recall-Kurven des Templatematching und des CNN auf der Seutter 1731 A.2.6 Karte gezeigt. Bei hohen Precision-Werten weist das Templatematching einen etwas höheren Recall auf. Dafür lässt sich mit dem CNN ein Recall von 0,937 erreichen wobei die Precision noch bei über 0,75 liegt, wohingegen sie beim Templatematching schon auf ca. 0,35 abgefallen ist.

3. Erkennung von Ortssignaturen

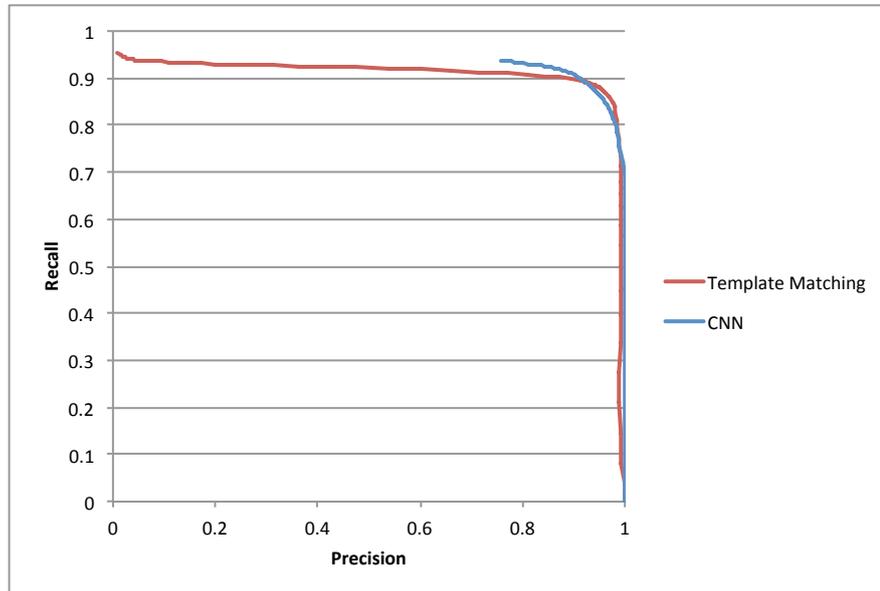


Abb. 3.17.: Vergleich der Precision-Recall-Kurven des Templatematching und des CNN auf der Seutter 1731 A.2.6 Karte.

Ergebnisse für die Rotenhan/Ortelius 1570 Karte

Folgende Ergebnisse wurden für die Rotenhan/Ortelius 1570 A.2.2 Karte erzielt.

Tab. 3.4.: Template-Matching Ortskringel Rotenhan/Ortelius 1570 auf Ausschnitt

Maske	Bester F1-Score	Schwellwert
A0	0,557377049	0,41
A1	0,390243902	0,54
A2	0,246575342	0,52
B0	0,677966102	0,68
B1	0,441176471	0,54
B2	0,271186441	0,55
T	0,136363636	0,64

In Tabelle 3.4 werden die Ergebnisse des Template-Matchings auf einem Ausschnitt der Rotenhan/Ortelius 1570 Karte A.2.2 gezeigt. Es sind jeweils für jede der sechs automatisch generierten Masken und für das volle Template der beste erreichbare F1-Score angegeben und der Schwellwert, mit dem der beste F1-Score erreicht wird. Der betrachtete Ausschnitt ist 1633x849 Pixel (1,32 MP) groß und enthält 28 Ortssignaturen. Mit dem Template lässt sich der beste F1-Score durch Verwendung

Tab. 3.5.: Template-Matching Ortskringel Rotenhan/Ortelius 1570

Maske	Bester F1-Score	Schwellwert
A0	0,377850163	0,41
A1	0,325138779	0,46
A2	0,221806854	0,5
B0	0,637596899	0,66
B1	0,44205298	0,53
B2	0,242384964	0,51
T	0,121900826	0,56

der B0 Maske mit einem Schwellwert von 0,68 erreichen. Diese Kombination aus Maske und Schwellwert wird nachfolgend auch verwendet um die Ortssignaturen in der gesamten Karte zu extrahieren.

Der annotierte Ausschnitt umfasst 5,59% der Kartenfläche von 4178x5925 Pixel (23,61 MP) und 5,10% der 549 Ortssignaturen. Schon mit dieser kleinen Menge an annotierten Orten lässt sich eine gute Kombination aus Maske und Schwellwert ermitteln, welche auf dem Gesamtbild einen F1-Score von 0,619506967 erreicht, was besser ist als alles, was sich mit den restlichen Masken erreichen lässt und nur 1,8089932 Prozentpunkte vom mit dieser Methode möglichen Optimum entfernt ist. In absoluten Zahlen bedeutet dies, es werden auf der Gesamtkarte 384 Orte markiert, wovon 289 richtig sind und 95 falsch. Dies entspricht bei 549 Orten einer Precision von 0,752604167 und einem Recall von 0,526411658. Das optimale Ergebnis liegt bei 554 markierten Orten, von denen 348 richtig und 206 falsch sind.

3. Erkennung von Ortssignaturen

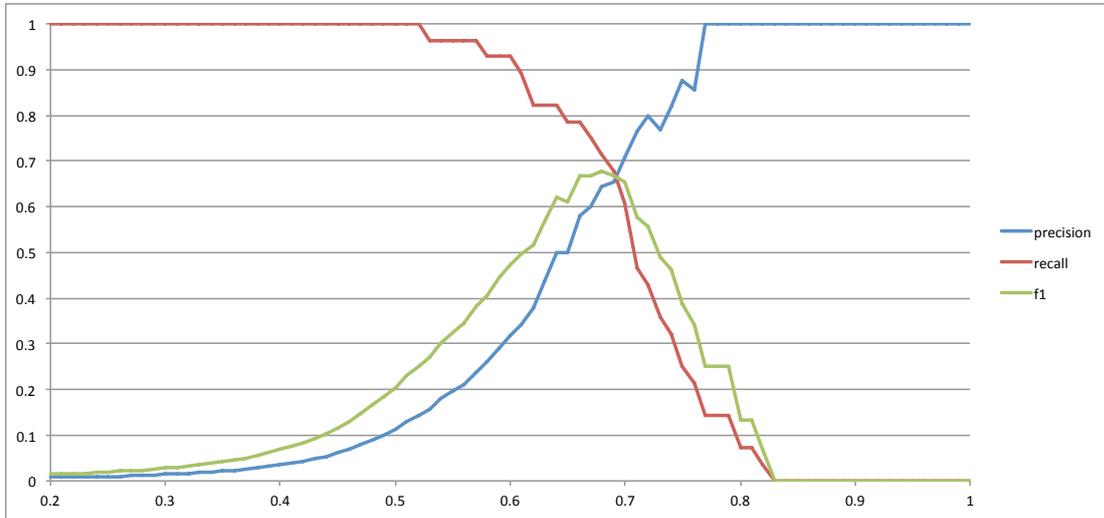


Abb. 3.18.: Auswertung des Templatematching mit Maske B0 auf einem Ausschnitt der Rotenhan/Ortelius 1570 A.2.1 Karte.

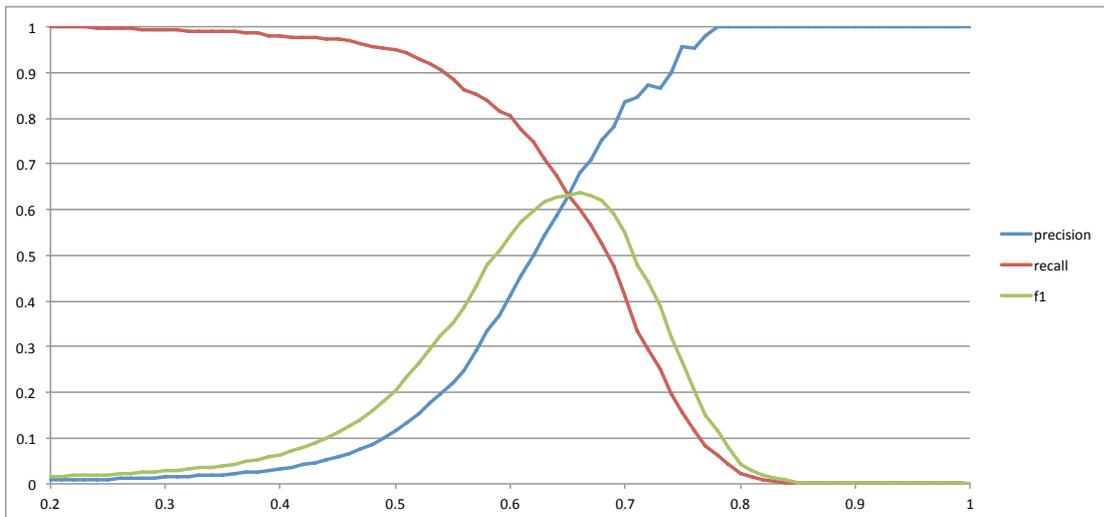


Abb. 3.19.: Auswertung des Templatematching für das Template mit Maske B0 auf der kompletten Rotenhan/Ortelius 1570 A.2.6 Karte.

Ergebnisse für die Merian 1656 Karte

Folgende Ergebnisse wurden für die Merian 1656 A.2.6 Karte erzielt.

In Tabelle 3.6 werden die Ergebnisse des Template-Matchings auf einem Aus-

Tab. 3.6.: Template-Matching Ortskringel Merian 1656 auf Ausschnitt

Maske	Bester F1-Score	Schwellwert
A0	0,738461538	0,61
A1	0,776119403	0,56
A2	0,774193548	0,61
B0	0,738461538	0,61
B1	0,776119403	0,56
B2	0,774193548	0,61
T	0,754098361	0,68

Tab. 3.7.: Template-Matching Ortskringel Merian 1656

Maske	Bester F1-Score	Schwellwert
A0	0,595673877	0,58
A1	0,639252336	0,54
A2	0,627254509	0,57
B0	0,595673877	0,58
B1	0,639252336	0,54
B2	0,627254509	0,57
T	0,598455598	0,66

schnitt der Merian 1656 Karte A.2.5 gezeigt. Es sind jeweils für jede der sechs automatisch generierten Masken und für das volle Template der beste erreichbare F1-Score angegeben und der Schwellwert, mit dem der beste F1-Score erreicht wird. Der betrachtete Ausschnitt ist 1600x1100 Pixel (1,68 MP) groß und enthält 37 Ortssignaturen. Mit dem Template lässt sich der beste F1-Score durch Verwendung der A1/B1 Maske mit einem Schwellwert von 0,54 erreichen. Diese Kombination aus Maske und Schwellwert wird nachfolgend auch verwendet um die Ortssignaturen in der gesamten Karte zu extrahieren. Die Masken der A- und B-Reihe sind hier identisch da die Masken der A-Reihe keine Löcher enthalten. 0,89017341 0,471669219 Der annotierte Ausschnitt umfasst 2,72% der Kartenfläche von 9067x7130 Pixel (61,65 MP) und 5,67% der 653 Ortssignaturen. Schon mit dieser kleinen Menge an annotierten Orten lässt sich eine gute Kombination aus Maske und Schwellwert ermitteln, welche auf dem Gesamtbild einen F1-Score von 0,616616617 erreicht, was sich durch die anderen Masken nur von den A2/B2 Masken knapp übertreffen lassen würde und nur 2,263572 Prozentpunkte vom mit dieser Methode möglichen Optimum entfernt ist. In absoluten Zahlen bedeutet dies, es werden auf der Gesamtkarte 346 Orte markiert, wovon 308 richtig sind

3. Erkennung von Ortssignaturen

und 36 falsch. Dies entspricht bei 653 Orten einer Precision von 0,89017341 und einem Recall von 0,471669219. Das optimale Ergebnis liegt bei 417 markierten Orten, von denen 342 richtig und 75 falsch sind.

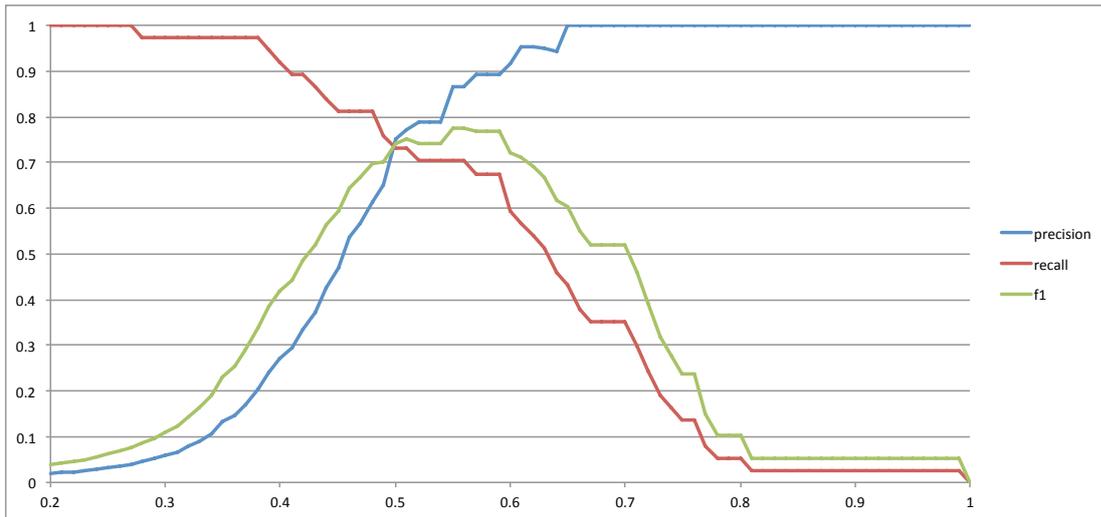


Abb. 3.20.: Auswertung des Templatematching mit Maske B1 auf einem Ausschnitt der Merian 1656 A.2.5 Karte.

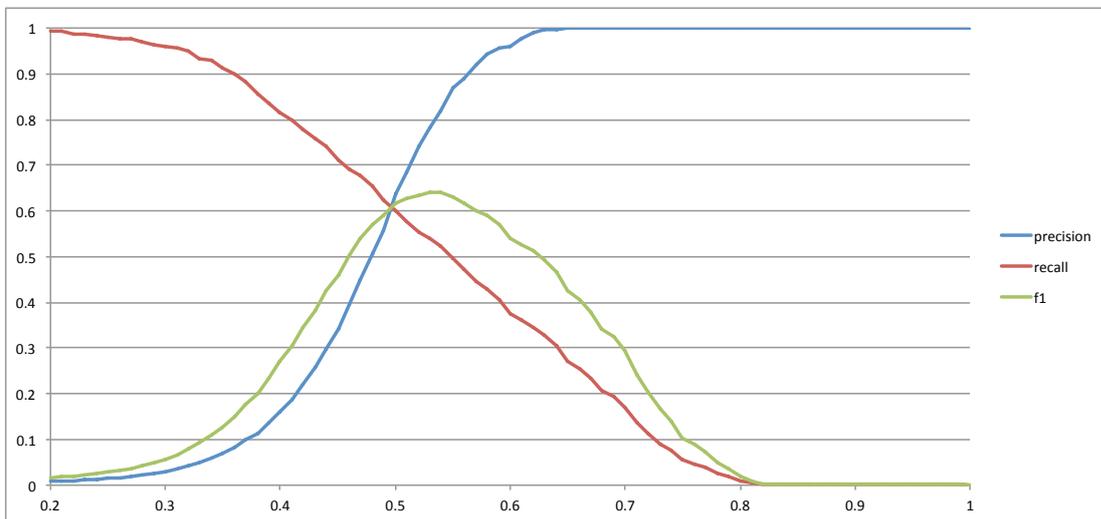


Abb. 3.21.: Auswertung des Templatematching für das Template mit Maske B1 auf der kompletten Merian 1656 A.2.5 Karte.

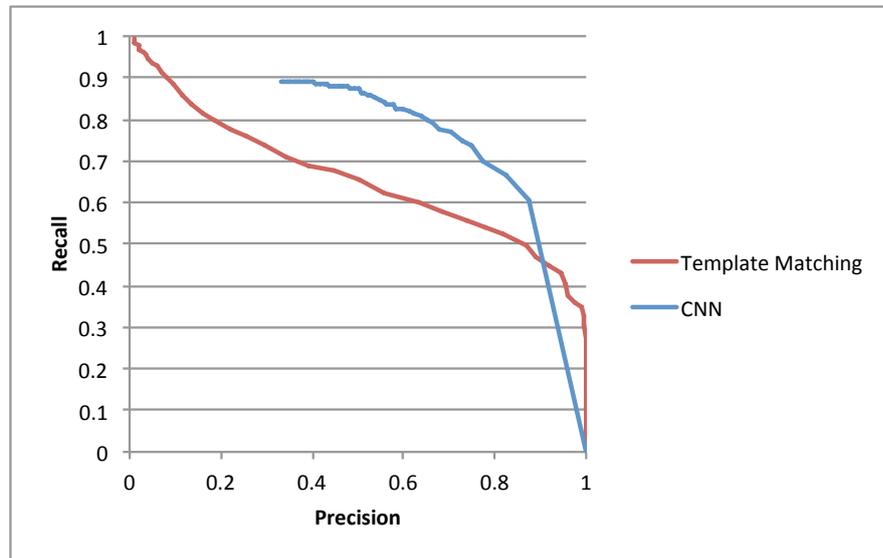


Abb. 3.22.: Vergleich der Precision-Recall-Kurven des Templatematching und des CNN auf der Merian 1656 A.2.5 Karte.

Ergebnisse für die Mercator 1600 Karte

Folgende Ergebnisse wurden für die Mercator 1600 A.2.3 Karte erzielt.

Tab. 3.8.: Template-Matching Ortskringel Mercator 1600 auf Ausschnitt

Maske	Bester F1-Score	Schwellwert
A0	0,972972973	0,49
A1	0,962566845	0,56
A2	0,91954023	0,64
B0	0,983606557	0,64
B1	0,972677596	0,61
B2	0,920454545	0,62
T	0,792207792	0,63

In Tabelle 3.8 werden die Ergebnisse des Template-Matchings auf einem Ausschnitt der Mercator 1600 Karte A.2.3 gezeigt. Es sind jeweils für jede der sechs automatisch generierten Masken und für das volle Template der beste erreichbare F1-Score angegeben und der Schwellwert, mit dem der beste F1-Score erreicht wird. Der betrachtete Ausschnitt ist 1638x905 Pixel (1,41 MP) groß und enthält 92 Ortssignaturen. Mit dem Template lässt sich der beste F1-Score durch Verwendung

3. Erkennung von Ortssignaturen

Tab. 3.9.: Template-Matching Ortskringel Mercator 1600

Maske	Bester F1-Score	Schwellwert
A0	0,954342984	0,52
A1	0,939732143	0,6
A2	0,903559127	0,62
B0	0,961453744	0,66
B1	0,943883277	0,63
B2	0,904458599	0,63
T	0,725707257	0,59

der B0 Maske mit einem Schwellwert von 0,64 erreichen. Diese Kombination aus Maske und Schwellwert wird nachfolgend auch verwendet um die Ortssignaturen in der gesamten Karte zu extrahieren.

Der annotierte Ausschnitt umfasst 5,77% der Kartenfläche von 6012x4260 Pixel (24,42 MP) und 10,03% der 917 Ortssignaturen. Schon mit dieser kleinen Menge an annotierten Orten lässt sich eine gute Kombination aus Maske und Schwellwert ermitteln, welche auf dem Gesamtbild einen F1-Score von 0,95156082 erreicht, was sich durch die anderen Masken nur von der A1 Maske knapp übertreffen lassen würde und nur 0,989292 Prozentpunkte vom mit dieser Methode möglichen Optimum entfernt ist. In absoluten Zahlen bedeutet dies, es werden auf der Gesamtkarte 941 Orte markiert, wovon 884 richtig sind und 57 falsch. Dies entspricht bei 917 Orten einer Precision von 0,93942614 und einem Recall von 0,96401309. Das optimale Ergebnis liegt bei 899 markierten Orten, von denen 873 richtig und 26 falsch sind.

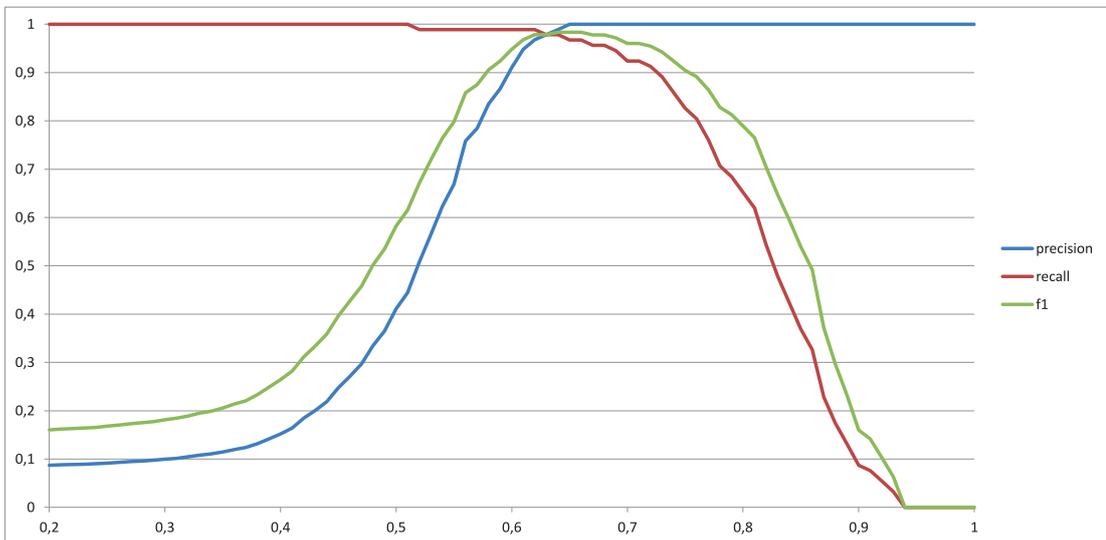


Abb. 3.23.: Auswertung des Templatematching mit Maske B0 auf einem Ausschnitt der Mercator 1600 A.2.3 Karte.

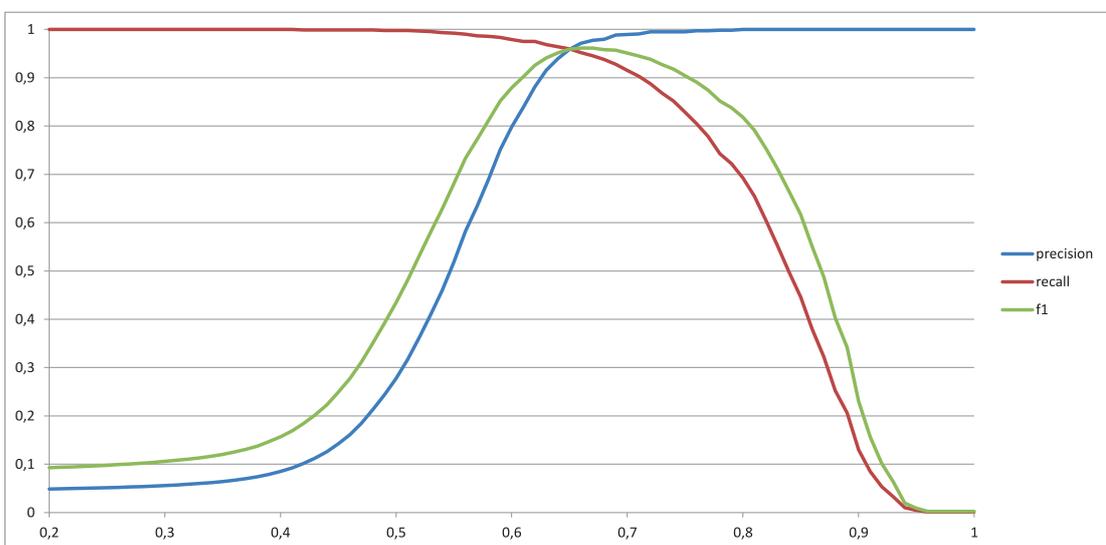


Abb. 3.24.: Auswertung des Templatematching für das Template mit Maske B0 auf der kompletten Mercator 1600 A.2.3 Karte.

Ergebnisse für die Ortelius 1612 Karte Z6

Folgende Ergebnisse wurden für die Ortelius 1612 A.1.5 Karte (Z6) erzielt.

3. Erkennung von Ortssignaturen

Tab. 3.10.: Template-Matching Ortskringel Ortelius 1612 auf Ausschnitt

Maske	Bester F1-Score	Schwellewert
A0	0,817518248	0,36
A1	0,830985915	0,5
A2	0,780141844	0,53
B0	0,772727273	0,7
B1	0,852941176	0,61
B2	0,786206897	0,55
T	0,756756757	0,54

Tab. 3.11.: Template-Matching Ortskringel Ortelius 1612

Maske	Bester F1-Score	Schwellewert
A0	0,688950789	0,36
A1	0,737027265	0,49
A2	0,682512733	0,53
B0	0,778837814	0,7
B1	0,758302583	0,59
B2	0,693733451	0,56
T	0,646253022	0,54

In Tabelle 3.10 werden die Ergebnisse des Template-Matchings auf einem Ausschnitt der Ortelius 1612 Karte A.1.5 gezeigt. Es sind jeweils für jede der sechs automatisch generierten Masken und für das volle Template der beste erreichbare F1-Score angegeben und der Schwellwert, mit dem der beste F1-Score erreicht wird. Der betrachtete Ausschnitt ist 1697×924 Pixel (1,5 MP) groß und enthält 68 Ortssignaturen. Mit dem Template lässt sich der beste F1-Score durch Verwendung der B1 Maske mit einem Schwellwert von 0,61 erreichen. Diese Kombination aus Maske und Schwellwert wird nachfolgend auch verwendet um die Ortssignaturen in der gesamten Karte zu extrahieren.

Der annotierte Ausschnitt umfasst 5,15% der Kartenfläche von 6221×4905 Pixel (29,1 MP) und 11,33% der 600 Ortssignaturen. Schon mit dieser kleinen Menge an annotierten Orten lässt sich eine gute Kombination aus Maske und Schwellwert ermitteln, welche auf dem Gesamtbild einen F1-Score von 0,74829932 erreicht, was sich nur mit der Maske B0 übertreffen lässt und nur 3,0538494 Prozentpunkte vom mit dieser Methode möglichen Optimum entfernt ist. In absoluten Zahlen bedeutet dies, es werden auf der Gesamtkarte 429 Orte markiert, wovon 385 richtig sind

und 44 falsch. Dies entspricht bei 600 Orten einer Precision von 0,897435897 und einem Recall von 0,641666667. Das optimale Ergebnis liegt bei 553 markierten Orten, von denen 449 richtig und 104 falsch sind.

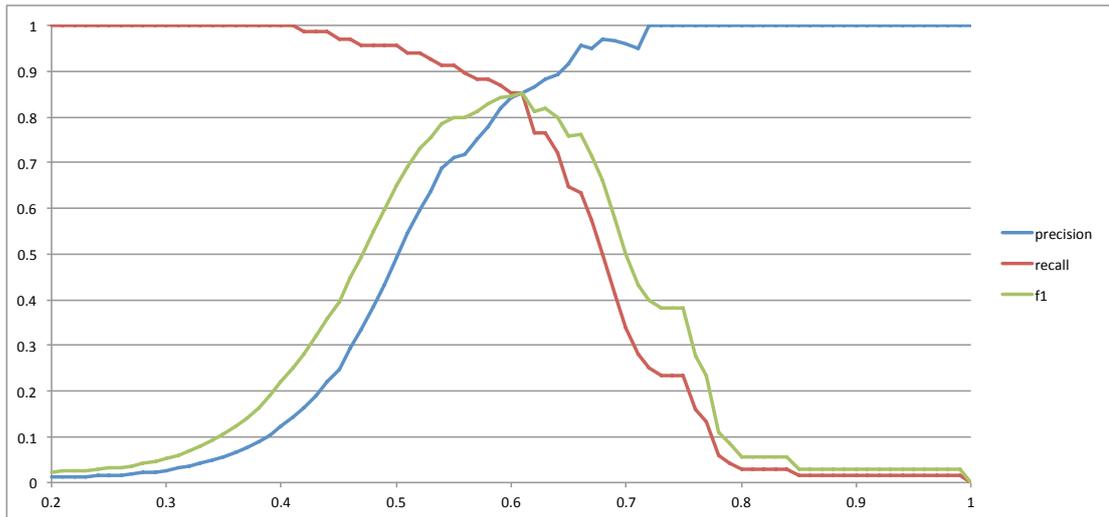


Abb. 3.25.: Auswertung des Templatematching mit Maske B1 auf einem Ausschnitt der Ortelius 1612 A.1.5 Karte.

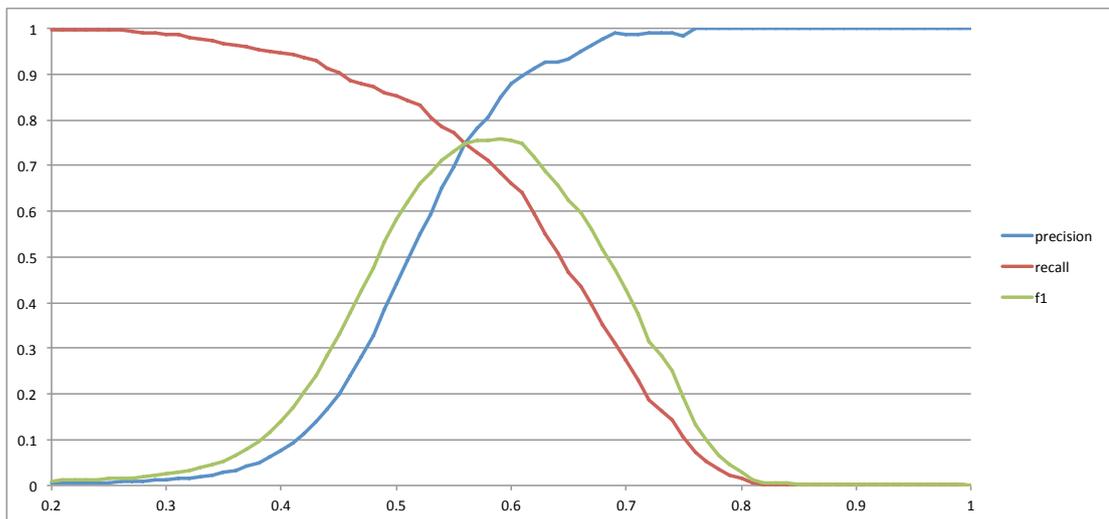


Abb. 3.26.: Auswertung des Templatematching für das Template mit Maske B1 auf der kompletten Ortelius 1612 A.1.5 Karte.

Abbildung 3.27 zeigt Precision und Recall für die Erkennung von Ortssignaturen mittels des in Abschnitt 3.3 vorgestellten Verfahrens auf der Ortelius 1612 A.1.5 Karte. Auch hier wurde der Kartenrand ausgeschlossen.

3. Erkennung von Ortssignaturen

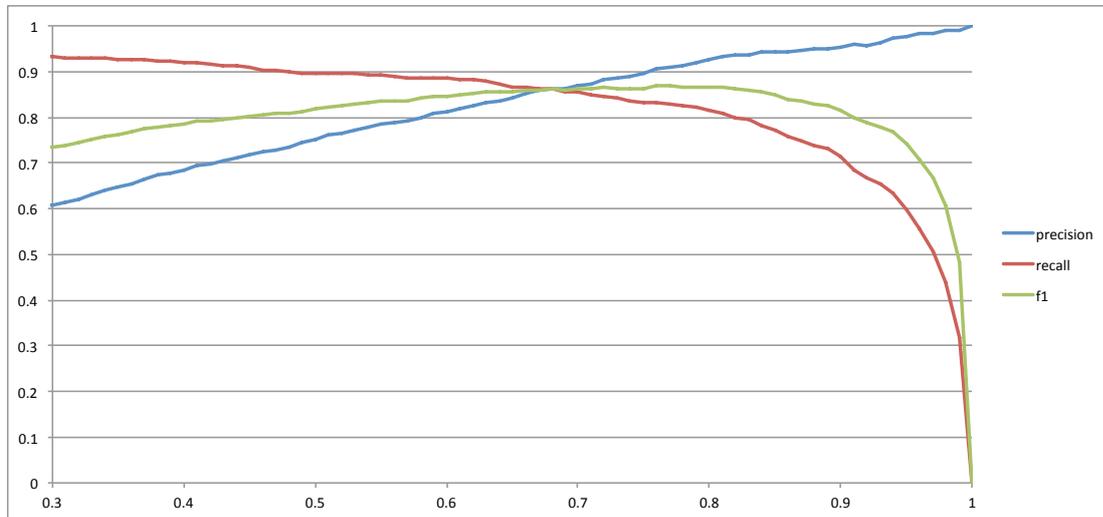


Abb. 3.27.: Auswertung der Erkennung von Ortssignaturen mittels CNN auf der Ortelius 1612 A.1.5 Karte.

In Abbildung 3.28 wird ein Vergleich der Precision-Recall-Kurven des Templatematching und des CNN auf der Ortelius 1612 A.1.5 Karte gezeigt. Bei Precision-Werten von über 0,985 weist das Templatematching einen höheren Recall auf, wobei auch dieser bei unter 0,5 liegt. Dafür der Recall des CNN für Precision-Werten von unter 0,985 deutlich über dem des Template-Matching.

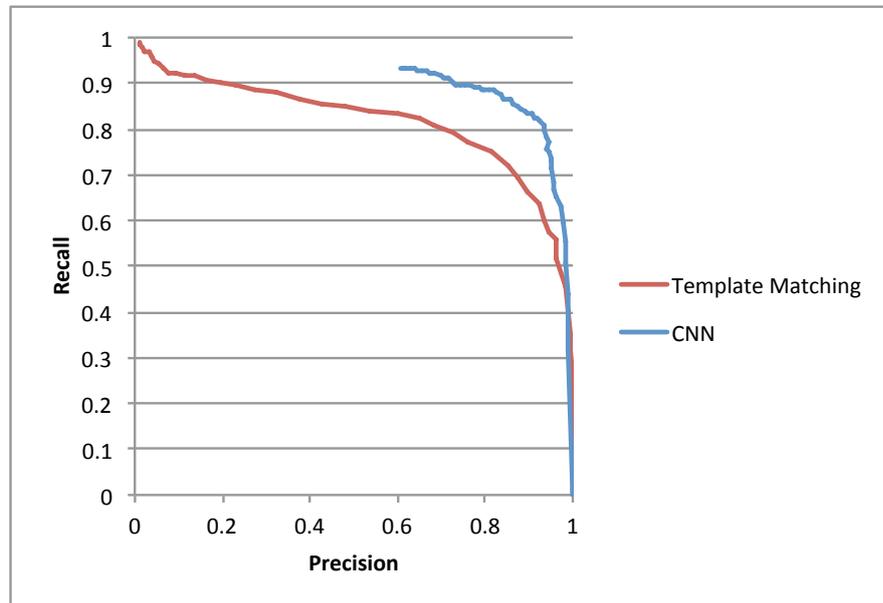


Abb. 3.28.: Vergleich der Precision-Recall-Kurven des Templatematching und des CNN auf der Ortelius 1612 A.1.5 Karte.

Ergebnisse für die Ortelius 1592 Karte Z5

Folgende Ergebnisse wurden für die Ortelius 1592 A.1.5 Karte (Z5) erzielt.

Tab. 3.12.: Template-Matching Ortskringel Ortelius 1592 auf Ausschnitt

Maske	Bester F1-Score	Schwellwert
A0	0,390804598	0,48
A1	0,696428571	0,46
A2	0,530612245	0,55
B0	0,76119403	0,65
B1	0,752	0,51
B2	0,530612245	0,56
T	0,452830189	0,56

In Tabelle 3.12 werden die Ergebnisse des Template-Matchings auf einem Ausschnitt der Ortelius 1592 Karte A.1.5 gezeigt. Es sind jeweils für jede der sechs automatisch generierten Masken und für das volle Template der beste erreichbare F1-Score angegeben und der Schwellwert, mit dem der beste F1-Score erreicht wird. Der betrachtete Ausschnitt ist 1697×924 Pixel (1,5 MP) groß und enthält 66

3. Erkennung von Ortssignaturen

Tab. 3.13.: Template-Matching Ortskringel Ortelius 1592

Maske	Bester F1-Score	Schwellwert
A0	0,381551363	0,46
A1	0,455223881	0,45
A2	0,321389794	0,53
B0	0,782459157	0,67
B1	0,554358472	0,52
B2	0,326304107	0,54
T	0,273944387	0,55

Ortssignaturen. Mit dem Template lässt sich der beste F1-Score durch Verwendung der B0 Maske mit einem Schwellwert von 0,65 erreichen. Diese Kombination aus Maske und Schwellwert wird nachfolgend auch verwendet um die Ortssignaturen in der gesamten Karte zu extrahieren.

Der annotierte Ausschnitt umfasst 4,66% der Kartenfläche von 6589×5103 Pixel (32,06 MP) und 11,04% der 598 Ortssignaturen. Schon mit dieser kleinen Menge an annotierten Orten lässt sich eine gute Kombination aus Maske und Schwellwert ermitteln, welche auf dem Gesamtbild einen F1-Score von 0,767899292 erreicht, was besser ist als alles, was sich mit den restlichen Masken erreichen lässt und nur 1,4559865 Prozentpunkte vom mit dieser Methode möglichen Optimum entfernt ist. In absoluten Zahlen bedeutet dies, es werden auf der Gesamtkarte 673 Orte markiert, wovon 488 richtig sind und 185 falsch. Dies entspricht bei 598 Orten einer Precision von 0,725111441 und einem Recall von 0,816053512. Das optimale Ergebnis liegt bei 565 markierten Orten, von denen 455 richtig und 110 falsch sind.

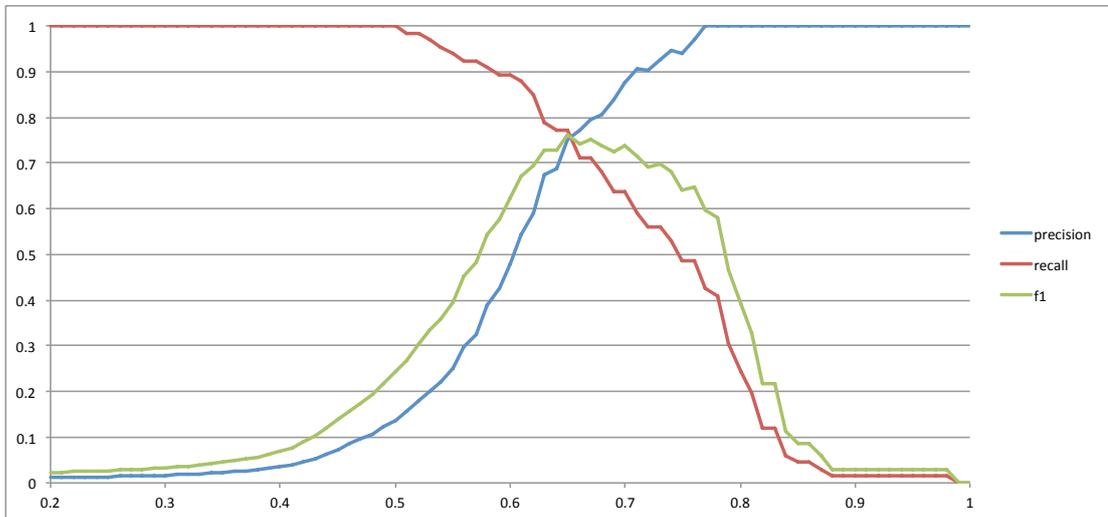


Abb. 3.29.: Auswertung des Templatematching mit Maske B0 auf einem Ausschnitt der Ortelius 1592 A.1.5 Karte.

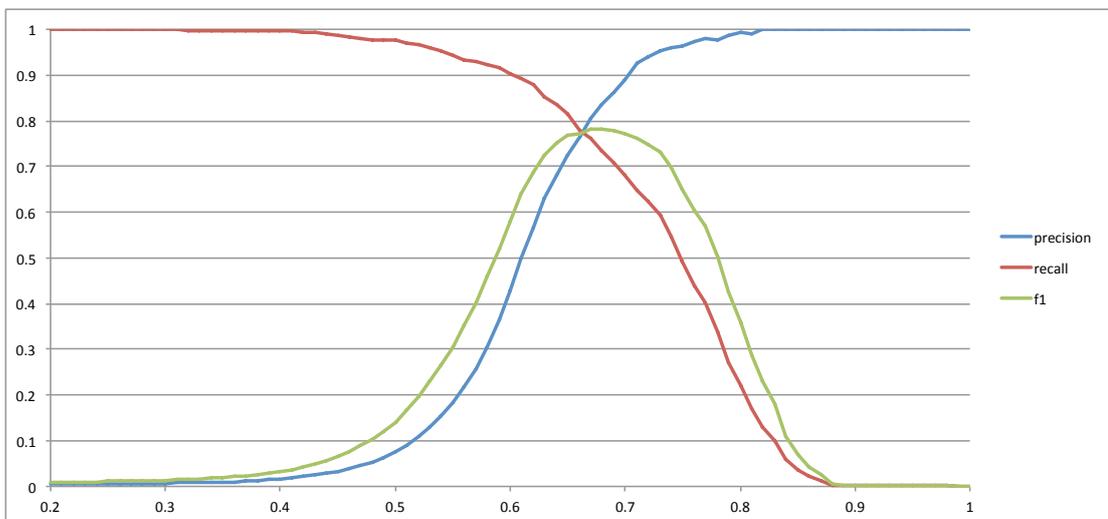


Abb. 3.30.: Auswertung des Templatematching für das Template mit Maske B0 auf der kompletten Ortelius 1592 A.1.5 Karte.

3.4.5. Diskussion der Ergebnisse

Beide in dieser Arbeit entwickelten Ansätze erreichen ähnliche Genauigkeiten, der F1-Score ist ähnlich und in jeweils der Hälfte der Fälle erreicht jede Methode den höchsten F1-Score. Beide neuen Ansätze sind in allen untersuchten Fällen dem

3. Erkennung von Ortssignaturen

einfachen Template-Matching überlegen. Das modifizierte Template-Matching hat beispielsweise in der Seutter 1731 Karte (A.2.6) in den Tests eine Precision von 98,2% bei einem Recall von 87,7% erreicht. Das CNN erreicht eine Precision von 93,2% bei einem Recall von 88,4%. Das bedeutet, es wurden mehr Ortssignaturen gefunden, in den Ergebnissen ist aber auch eine etwas höhere Zahl von Fehlern. Für die manuelle Nachkorrektur ist es aber einfacher, die fehlerhaften Treffer auszusortieren als fehlende Treffer in der Karte zu identifizieren. Zusätzlich lässt sich beim CNN die Erkennungsrate noch steigern ohne extreme Einbrüche bei der Erkennungsgenauigkeit zu erleiden. Im CNN werden bisher nur Convolutional-Layer verwendet und auch keine Erweiterungen des Trainings, wie z.B. Data-Augmentation oder Pre-Training, die in fast allen Fällen, besonders bei kleinen Trainingssets, zu einer deutlichen Performancesteigerung führen. Für das CNN besteht auch die Möglichkeit, falls genügend annotierte Daten von verschiedenen Karten vorhanden sind, ein Modell zu trainieren, das Ortssignaturen auch auf unbekanntem Karten erkennt. Somit zeigt der auf CNNs basierende Ansatz deutlich mehr Potential für zukünftige Leistungssteigerungen.

3.5. Zusammenfassung

Dieses Kapitel hat die möglichen Lösungen vorgestellt, um die erste Forschungsfrage zu beantworten:

Lassen sich Ortssignaturen in Altkarten automatisch erkennen?

Es wurden zwei Lösungsansätze vorgestellt. Insgesamt wurden die folgenden Techniken analysiert:

1. Modifiziertes Template-Matching
2. Convolutional-Neural-Network

Die Experimente haben gezeigt dass beide Ansätze eine ähnliche Genauigkeit erreichen, welche über der des einfachen Template-Matching liegt. Der auf CNNs basierende Ansatz zeigt jedoch deutlich mehr Potential für zukünftige Leistungssteigerungen.

4. Erkennung der Beschriftung von Ortssignaturen

Dieses Kapitel behandelt die im Abschnitt 1.2 eingeführte zweite Forschungsfrage im Bereich der Erkennung von Ortslabels. Diese Arbeit baut auf den im Abschnitt 2.2 besprochenen zwei Schritten wie folgt auf:

1. zunächst wird die Textebene in einer Karte von allen anderen Ebenen getrennt, auf diese Weise werden die Textbereiche erkannt;
2. anschließend werden die Texte in den identifizierten Textbereichen automatisch lesbar gemacht.

Abschnitt 4.1 beschreibt ausführlich den Prozess der Erkennung der Textbereiche in historischen Karten. Dazu werden zwei unterschiedliche Methoden getestet, die auf unterschiedlichen Modellen aufbauen. Im Abschnitt 4.2 werden zwei unterschiedliche Ansätze zur Texterkennung erklärt und deren Erkennungsgenauigkeit evaluiert. Die Gesamtergebnisse werden im Abschnitt 4.3 diskutiert.

4.1. Erkennung von Textbereichen

Karten bestehen aus verschiedenen Ebenen mit verschiedenen Informationstypen, z. B. Transportwegen, Grenzen, Hydrographie und geographische Orte. Die geographischen Orte werden häufig mit unterschiedlichen Informationen angegeben, wie zum Beispiel Ortsposition, Ortstyp, Ortsgröße und Ortsname. Für die automatische Analyse ist es nötig, die Karte in einzelne Ebenen zu zerlegen, welche dann mit

4. Erkennung der Beschriftung von Ortssignaturen

verschiedenen Methoden zur Extraktion von Informationen verarbeitet werden. Zur Extraktion von textuellen Informationen (z. B. Namen geographischer Orte verschiedener Typen und Größen) können Texterkennungsmethoden angewendet werden.



Abb. 4.1.: Ein Ausschnitt aus der Altkarte, Circulus Franconicus von Matthias Seutter (1731) A.2.6

Ähnlich wie für die automatische Identifikation von Kartensymbolen gelten für die Erkennung von Textlabels die gleichen harten Randbedingungen. Generell lässt sich die Aufgabe als Identifikation von Text in Altkarten mit großer Variation in Stil, Größe und Ausrichtung beschreiben. Ein tieferer Blick stellt folgende diese Aufgabe erschwerenden Charakteristiken fest:

1. Im Gegensatz zur Textsegmentierung in modernen Karten ist die Segmentierung der Textebene in Altkarten durch Alterungseffekte erschwert, wie z. B. vergilbtes Papier oder ausgebleichene Regionen.
2. Zusätzlich hat der Text keine regelmäßige Zeilenstruktur, er kann beliebige Ausrichtung haben oder auch gekrümmt sein.

3. Außerdem enthalten Altkarten auch Symbole für Bäume und Ortsmarkierungen, welche in den Größenbereich von Text fallen. Solche Symbole werden im Rest dieser Arbeit als *nicht-textuelle Symbole* bezeichnet.

Abbildung 4.1 zeigt einen typischen Ausschnitt einer Altkarte und macht deutlich, wie die oben genannten Herausforderungen aussehen können. Speziell die nicht-textuellen Symbole sind hier reichlich vorhanden.

Um gegen diese Herausforderungen anzutreten, werden zwei Verfahren zur Erkennung der Textbereiche untersucht. Wie im Abschnitt 2.2.1 motiviert, werden zwei prinzipiell unterschiedliche Verfahren zur Textlokalisierung an die Domäne der Altkarten angepasst und miteinander verglichen:

1. In Abschnitt 4.1.1 wird ein klassisches regelbasiertes Verfahren zur Trennung von Text und Grafiken (Tombre et al., 2002) modifiziert um es besser an die Eigenschaften von Altkarten anzupassen.
2. Anschließend wird in Abschnitt 4.1.2 ein Deep-Learning-basiertes Verfahren (Zhou et al., 2017) getestet und durch zusätzliches Training an die Eigenschaften der Altkarten angepasst.

Die Ergebnisse werden im Abschnitt 4.1.3 vorgestellt und diskutiert.

4.1.1. Modifiziertes regelbasiertes Verfahren zur Textlokalisierung in Altkarten

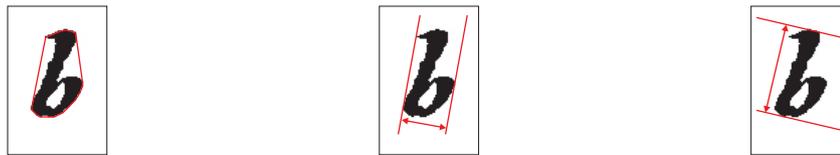
Teile dieses Abschnitts wurden im Konferenzartikel (Höhn, 2013) auf der Iberian Conference on Pattern Recognition and Image Analysis 2013 publiziert.

Das Verfahren basiert auf dem Verfahren von Tombre et al. (2002), welches auf einer Filterung von Zusammenhangskomponenten beruht. Die Zusammenhangskomponenten eines Binärbildes erhält man, indem man alle schwarzen Pixel, die über Wege aus schwarzen Pixeln erreichbar sind, zusammen gruppiert. Diese Zusammenhangskomponenten werden nach ihrem Verhältnis von Höhe und Breite, sowie ihrer Fläche in Text und Grafikkomponenten aufgeteilt. Aus den so kategorisierten

4. Erkennung der Beschriftung von Ortssignaturen

Textkomponenten werden dann noch lange schmale Komponenten entfernt um die endgültige Menge an Textkomponenten zu erhalten.

Diese Arbeit verwendet auch eine Analyse der Zusammenhangskomponenten für die Trennung von Text und Grafik, hierbei werden die Dichte sowie minimale und maximale Durchmesser als Hauptfeatures benutzt. Für eine Zusammenhangskomponente c ist die Dichte $\text{dens}(c)$ als Verhältnis zwischen der Fläche der konvexen Hülle und der Anzahl an schwarzen Pixeln definiert. Die Dichte wird auf 1 gesetzt, falls $\text{dens}(c) < 1$ ist. Aus dem minimalen Durchmesser $\varnothing_{\min}(c)$ und dem maximalen Durchmesser $\varnothing_{\max}(c)$ lässt sich das Durchmesser Verhältnis $\text{dr}(c) = \varnothing_{\max}(c)/\varnothing_{\min}(c)$ berechnen. Beispiele für die konvexe Hülle und den minimalen und maximalen Durchmesser werden in Abbildung 4.2 gezeigt. Die Dichte und das Durchmesser Verhältnis sind skalierungs- und rotationsinvariant und dadurch unabhängig von der Textrichtung oder Bildauflösung.



(a) Konvexe Hülle (b) Minimaler Durchmesser (c) Maximaler Durchmesser

Abb. 4.2.: Veranschaulichung von konvexer Hülle, minimalem Durchmesser und maximalem Durchmesser.

Zusätzlich wird noch die Definition für die kürzeste Entfernung zwischen zwei Zusammenhangskomponenten $\text{dist}(c_1, c_2)$ benötigt, welche die kürzeste Entfernung zwischen den Pixeln in den zwei Zusammenhangskomponenten angibt. Durch die starken Größenvariationen von Text- und Nicht-Text-Komponenten können wir keinen Schwellwert zur Trennung angeben, wie er bei der Text/Grafik-Trennung verwendet wird. Daher benutzen wir einen Feature-Vektor mit einer Schwellwertregion für Text und kombinieren dies mit einer Analyse von benachbarten Komponenten um Nicht-Text-Komponenten auszufiltern.

Insgesamt besteht die Methode aus den folgenden Schritten:

1. Entfernen von gepunkteten und gestrichelten Linien,
2. Bestimmung der x -Höhe,

3. Filtern der Text-Komponenten,
4. Clustering der Text-Komponenten,
5. Finale Auswahl der Komponenten.

Die folgenden Unterabschnitte erklären die Methode Schritt für Schritt. Als Ausgangspunkt für diese Schritte wird die Menge aller Zusammenhangskomponenten (CC_{all}) aus dem binarisierten Bild verwendet.

4.1.1.1. Entfernen von gepunkteten und gestrichelten Linien

Für die spätere Schätzung der x -Höhe betrachten wir benachbarte Kandidaten für die Text-Komponenten. Gepunktete und gestrichelte Linien können diesen Schritt beeinflussen, die Linienstärke könnte mit der x -Höhe verwechselt werden. Daher werden die gestrichelten und gepunkteten Linien in einem Vorverarbeitungsschritt entfernt. Dies geschieht wie folgt.

Die Zusammenhangskomponenten aus CC_{all} werden nach ihrer Dichte gefiltert. Alle Komponenten c mit $\text{dens}(c) < 1.2$ werden zur Menge der möglichen Liniensegmente (PLC) hinzugefügt. Durch diesen Filter bekommt man nur nahezu konvexe Objekte, aus denen man dann eine Menge von benachbarten Objekten mit ähnlichen Größen erzeugen kann. Ein Paar wird als ähnlich groß betrachtet, wenn $1/1.2 \leq \varnothing_{\min}(c_1)/\varnothing_{\min}(c_2) \leq 1.2$ gilt. Als benachbart wird es betrachtet wenn $\text{dist}(c_1, c_2) \leq 2 \cdot \max(\varnothing_{\max}(c_1), \varnothing_{\max}(c_2))$ gilt.

Beim Größenvergleich hilft der minimale Durchmesser, da die einzelnen Liniensegmente auch verschiedene Längen haben können, die Strichstärke aber ungefähr konstant ist. Sobald wir drei oder mehr benachbarte und ähnlich große Liniensegmente aus PLC finden, werden diese Komponenten zur Menge der Liniensegmente CC_{dash} hinzugefügt.

Diese Menge ist von der Bestimmung der x -Höhe ausgeschlossen aber nicht von den darauffolgenden Schritten, wodurch Fehler in diesem Schritt nicht die spätere Erkennung als Textkomponente verhindern.

4. Erkennung der Beschriftung von Ortssignaturen

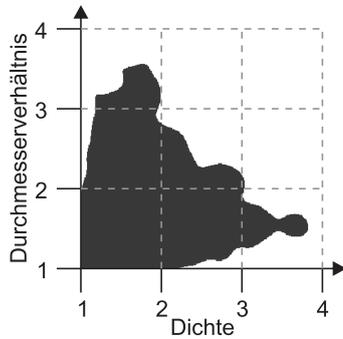


Abb. 4.3.: Schwellwertbereich

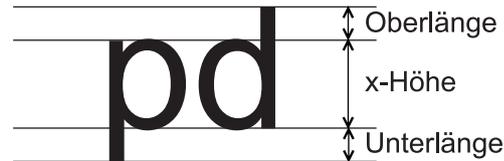


Abb. 4.4.: Typografische Linien

4.1.1.2. Bestimmung der x -Höhe

Um Zusammenhangskomponenten zur Bestimmung der x -Höhe auszuwählen, werden die Dichte und das Durchmesser Verhältnis benutzt. Dazu wird für die Zusammenhangskomponenten bestimmt, ob der Punkt, der aus Dichte und Durchmesser Verhältnis gebildet wird, in dem in Abb. 4.3 dargestellten Schwellwertbereich liegt.

Um stabile Schwellwerte zu erhalten, wurden 1214 Schriftarten analysiert. Die Schriftarten wurden von “Google web fonts” (Google Fonts, 2018) und “Fraktur mon Amour” (Schalansky, 2008) gesammelt, da viele der Karten Fraktur verwenden. In dieser Sammlung kommen aber auch künstlerische Schriften, Schreibschriften und Schriftarten für Schriftsysteme außer dem Lateinischen vor. Daher wurden nur Schriftarten in Betracht gezogen, bei denen im gerenderten Text die Anzahl der Zusammenhangskomponenten um weniger als 20% von der Zeichenzahl des Textes abweicht, und alle im Text verwendeten Buchstaben vorhanden waren. Die Teststrings bestanden aus tausend zufällig gewählten deutschen Ortsnamen und wurden in verschiedenen Größen gerendert. Für jede Zusammenhangskomponente wurden die Dichte und das Durchmesser Verhältnis berechnet und damit ein zweidimensionales Histogramm gebildet. Die Fläche in diesem Histogramm über 0.1% vom Maximalwert wurde zur Trennung von Text und Nicht-Text gewählt.

Wenn dieser Filter auf die Zusammenhangskomponenten CC_{all} ohne CC_{dash} angewendet wird, bekommt man die Menge CC_{ta} , die hauptsächlich aus Einzelzeichen und wenigen verbundenen Zeichen besteht.

In der Menge CC_{ta} wird nun nach benachbarten Zusammenhangskomponenten gesucht. Dazu wird das Intervall von der kleinsten zur größten x -Höhe bestimmt, für welche die Komponenten mindestens den Bereich zwischen der Grund- und x -Linie ausfüllen, aber nicht den Bereich der Unter- und Oberlänge verlassen und zusätzlich einen kleineren Abstand als die x -Höhe haben. Der Maximalwert für die Oberlänge ar_{\max} und Unterlänge dr_{\max} wird relativ zu der x -Höhe angegeben um unabhängig von der Buchstabengröße zu bleiben. Als Ergebnis erhält man eine neue Menge von benachbarten Zusammenhangskomponenten.

Aus den Intervallen für die x -Höhe der gefundenen benachbarten Zusammenhangskomponenten bildet man ein Histogramm. Die x -Höhe wird mit den meisten Stimmen gewählt. Dies ist die geschätzte x -Höhe h_{est} .

4.1.1.3. Filtern der Text-Komponenten

Dieser Schritt bestimmt aus der Menge CC_{all} die Zusammenhangskomponenten, die mit der vorher berechneten x -Höhe und der maximalen Zeilenhöhe kompatibel sind. Die maximale Zeilenhöhe ist definiert als:

$$lh_{\max} = (1 + ar_{\max} + dr_{\max}) \cdot h_{\text{est}} \quad (4.1)$$

Eine Zusammenhangskomponente c wird als kompatibel angesehen, wenn $\varnothing_{\min}(c) \leq lh_{\max}$ gilt und zusätzlich mindestens eine der folgenden Bedingungen erfüllt ist:

$$\varnothing_{\min}(c) \geq h_{\text{est}} \quad (4.2)$$

$$h_{\text{est}} \leq \varnothing_{\max}(c) \leq lh_{\max} \cdot \alpha \quad (4.3)$$

Kursive und sich berührende Zeichen haben einen größeren maximalen Durchmesser als lh_{\max} und passen dennoch in die Zeile. Als Ausgleich dafür benutzen wir den Faktor α , welcher in unseren Berechnungen auf $\sqrt{2}$ gesetzt ist. Dies ist ausreichend um eine Schrägstellung um 45° auszugleichen und in kursiven Schriften wird gewöhnlich mit weniger starken Schrägstellungen gearbeitet. Als Bezeichnung für die Ergebnismenge dieses Schrittes verwenden wir im Folgenden FTC .

4. Erkennung der Beschriftung von Ortssignaturen

4.1.1.4. Clustering der Text-Komponenten

In diesem Schritt werden die Zusammenhangskomponenten zu Zeilen zusammengefasst. Dazu werden alle geordneten Paare von Komponenten aus der Menge FTC getestet, ob ihr Abstand kleiner als $h_{est}/2$ ist und sie potentiell eine Zeile bilden können. Zwei oder mehr Komponenten können potentiell eine Zeile mit der x -Höhe h_{est} bilden, wenn die Komponenten jeweils mindestens den Bereich zwischen der Grund- und x -Linie ausfüllen aber nicht den Bereich der Unter- und Oberlänge verlassen (s. auch Abbildung 4.4 zur Veranschaulichung). Man erzeugt eine Startmenge aus allen Paaren, die diese Bedingungen erfüllen, wobei die Paare geordnete Tupel sind.

Von allen Tupeln mit einer gemeinsamen Komponente c an der ersten Stelle werden nur die Tupel als Kandidaten behalten, wenn sie zwei nächste Nachbarn zu c enthalten und deren inverse Tupel. Diese Menge wird als NTC bezeichnet. Nach diesem Schritt kann eine Komponente trotzdem mehr als zwei Nachbarn haben.

Nun bildet man aus den Komponenten zusammen mit den Tupeln aus NTC als Startpunkt iterativ Zeilen. In jeder Iteration erweitert man die Enden der Tupel aus dem vorherigen Schritt um ihre benachbarten Komponenten. Wenn sie dann immer noch die Bedingungen für eine Zeile erfüllen, werden sie zum Ergebnis dieses Schrittes hinzugenommen.

Die Bedingungen für eine Zeile sind die gleichen wie im Fall von zwei Komponenten beschrieben. Hierdurch erhält man in jedem Schritt Zeilen mit einer zusätzlichen Komponente. Die Ergebnisse der Einzelschritte kombiniert man zu der Menge der möglichen Zeilen. Diese Zeilen können sich überlappen oder auch ineinander enthalten sein. Da im Folgenden die Anzahl der Komponenten in einer Zeile für die finale Auswahl von Textkomponenten benutzt wird, werden weiterhin Zeilen vereinigt, welche sich an ihren Enden um mindestens drei Komponenten überlappen zu neuen Zeilen ohne weitere Prüfung von Bedingungen. Dadurch können auch Zeilen mit leichten Biegungen zusammengefasst werden.

4.1.1.5. Finale Auswahl der Komponenten

Zum Abschluss werden die gefundenen Zeilen nach ihrer Komponentenanzahl sortiert und zur Ergebnismenge hinzugefügt. Diese Menge wird hier CC_{final} genannt. Beginnend mit der längsten Zeile werden alle ihre Komponenten zu CC_{final} hinzugefügt. Alle weiteren Zeilen, die eine dieser Komponenten enthalten, werden entfernt. Dies wird wiederholt, bis alle Zeilen bearbeitet sind. Die Komponenten in der Menge CC_{final} entsprechen nun den von der Methode als Textkomponente erkannten Komponenten.

4.1.2. Deep Learning zur Textlokalisierung in Altkarten

Obwohl die Erkennung von Text in Bildern und Fotografien in den letzten Jahren eine beeindruckende Performancesteigerung erreicht hat (siehe Abbildung 4.5), lassen sich viele Ansätze nicht direkt auf Altkarten übertragen, da sie oft auf die Eigenheiten der großen Wettbewerbe und der dort verwendeten Datensätze (Veit et al., 2016) angelegt sind. Convolutional-Neural-Networks bieten auch hier einen vielversprechenden Ansatz. Die wichtigste Einschränkung entsteht dadurch, dass meist achsenparallele Rechtecke für die Erkennung verwendet werden und die Beschriftungen in den Wettbewerben fast ausschließlich eine horizontale Ausrichtung aufweisen. Hinzu kommt ein weiteres prinzipielles Problem, wenn direkt die Bounding-Box eines Textes bestimmt werden soll. Texte weisen ein hohes Seitenverhältnis von Länge zu Höhe auf, was auch ein großes Sichtfeld für das CNN impliziert, da es sowohl Start als auch Ende des Textes sehen muss, um die Bounding-Box korrekt zu bestimmen. Für horizontale Texte lässt sich dieses Problem abmildern, indem mehrere den typischen Seitenverhältnissen von Text angepasste Sichtfelder für das CNN verwendet werden.

Es werden immer mehr Ansätze vorgestellt, die auch rotierten Text erkennen sollen. Je weniger zusätzliche Filter und Verarbeitungsschritte diese Ansätze aufweisen, desto höher ist die Wahrscheinlichkeit, dass sie bei einem Training auf Altkartendaten sich gut an diese anpassen lassen. EAST, ein Ansatz der nur aus zwei Stufen besteht wird von Zhou et al. (2017) beschrieben. Die erste Stufe besteht aus einem Fully Convolutional Network (FCN), welches eine pixelweise Vorhersage für

4. Erkennung der Beschriftung von Ortssignaturen

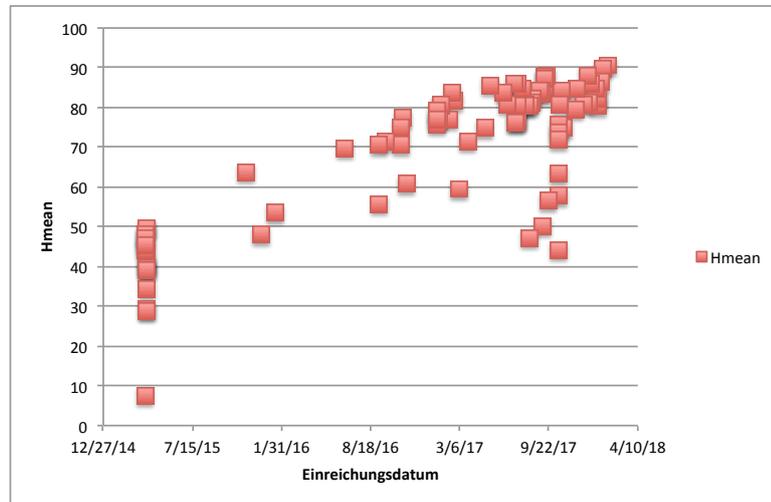


Abb. 4.5.: Ergebnisentwicklung bei der ICDAR Robust Reading Competition - Incidental Scene Text - Task 1

die Textregionen produziert. Die zweite Stufe ist ähnlich zu einer *non-maximum supression* (NMS), welche bei sich stark überschneidenden Vorhersagen nur diejenige mit dem höchsten Score beibehält und alle anderen ausfiltert. In EAST wird der Ansatz dahingehend modifiziert, dass keine Vorhersagen ausgefiltert werden, sondern die Vorhersagen werden durch ihre gewichteten Mittelwerte ersetzt. Dieses Vorgehen wird als *Locality-Aware NMS* bezeichnet. Die erste Stufe verwendet eine U-Net-Architektur (Ronneberger et al., 2015), um pixelweise vorherzusagen, wo sich Text befindet und auch pixelweise eine Vorhersage für die Bounding-Box des Textes zu treffen, siehe Abbildung 4.6. Die Bounding-Box wird durch die Angabe des Abstandes vom entsprechenden Pixel zum oberen, rechten, unteren und linken Rand zusammen mit dem Rotationswinkel der entsprechenden Bounding-Box beschrieben.

Eine Nachimplementierung von Zhou et al. (2017) in TensorFlow ist auf GitHub (argman, 2018) verfügbar. Diese Implementierung weist einige Abweichungen von der in Zhou et al. (2017) beschriebenen Implementierung auf, erreicht aber einen geringfügig besseren F1-Score im ICDAR 2015 Incidental Scene Text Detection Benchmark von 0,8083 im Vergleich zu 0,7820. Die Hauptunterschiede der Nachimplementierung sind das Ersetzen von PVANET (Kim et al., 2016) durch ResNet-50 (He et al., 2015) und die Verwendung des *dice loss* statt der *balanced cross entropy* als Kostenfunktion für die Vorhersage der Textpixel.

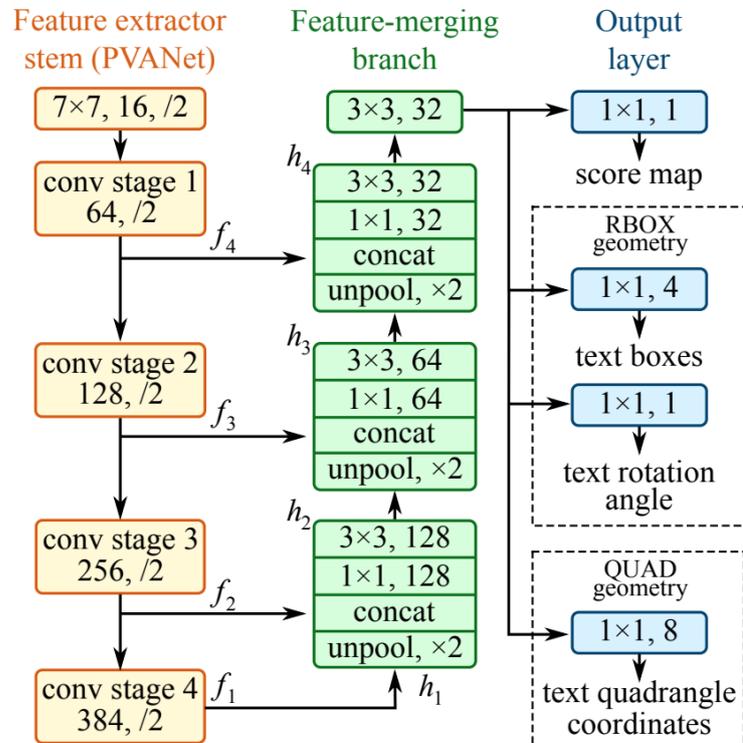


Abb. 4.6.: Architektur der ersten Stufe von EAST. Abbildung aus (Zhou et al., 2017)

Ein auf den Daten der ICDAR 2015 und 2013 Incidental Scene Text Detection Challenge trainiertes Modell¹ steht zum Download zur Verfügung und es ist auch möglich ein eigenes Modell zu trainieren oder ein Feintuning eines bestehenden Modells durchzuführen. Für das Training werden die Bilder mit entsprechenden Ground-Truth-Dateien benötigt. Die Ground-Truth-Dateien müssen für jedes im Bild vorkommende Textlabel die Koordinaten der vier Ecken der Bounding-Box im Uhrzeigersinn enthalten, gefolgt von dem transkribierten Text. Alle Elemente der Ground-Truth-Datei werden durch Kommas getrennt.

¹<https://drive.google.com/open?id=0B3APw5BZJ67ETHNPauU9xUkVoV0U> (zuletzt abgerufen am 31.07.2017)

4. Erkennung der Beschriftung von Ortssignaturen

4.1.3. Evaluierung

Die Erweiterung des klassischen Ansatzes und der Deep-Learning-Ansatz, mit und ohne zusätzliches Training für die Kartendaten werden auf einer Altkarte getestet.

4.1.3.1. Experimentaufbau

Die Ansätze wurden auf einer Altkarte mit 929 Ortslabels ausgewertet. Alle Ortslabels wurden mit Bounding-Boxen annotiert, welche an der Grundlinie und x-Höhe ausgerichtet sind. Dies lässt eine konsistentere Ausrichtung der Texte in den ausgeschnittenen Ortslabels zu. Es wird getestet, wie hoch die Erkennungsrate für die Ortslabels ist und wie viele fehlerhafte Erkennungen auftreten.

4.1.3.2. Evaluierungsmethodik

Zur Evaluierung wird das Intersection-over-Union (IoU) Kriterium benutzt, dies misst das Verhältnis der überlappenden Fläche zweier Rechtecke und der kombinierten Fläche. Dieses Kriterium wird auch im Incidental-Scene-Text-Task der Robust Reading Competition verwendet. Task 4.1: Text Localisation of Incidental Scene Text unterscheidet zwischen lesbaren Wörtern in lateinischer Schrift und unlesbaren Wörtern oder Wörtern in einem anderen Alphabet. Die lesbaren Wörtern in lateinischer Schrift müssen erkannt werden und die anderen Wörtern werden für die Auswertung ignoriert. Ein IoU-Wert von über 0,5 wird als Übereinstimmung gewertet. Zuerst werden Boxen, die mit ignorierten Ground-Truth-Boxen übereinstimmen, entfernt und dann die Recall- und Precision-Werte für die nicht ignorierten Boxen berechnet. Der F1-Score, das harmonische Mittel aus Precision und Recall, wird als kombinierter Score zur Bewertung der Methode verwendet. Auf der Webseite der Robust Reading Competition² werden zusätzlich zur möglichen Onlineauswertung Skripte für eine Offlineauswertung zur Verfügung gestellt. Diese Skripte werden auch hier zur Auswertung verwendet. (Karatzas et al., 2015)

²<http://rrc.cvc.uab.es/?ch=4&com=mymethods&task=1> (zuletzt abgerufen am 31.07.2017)

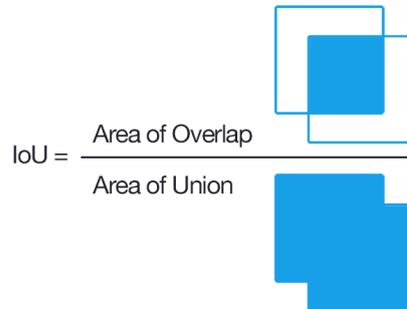


Abb. 4.7.: Intersection over Union

Das IoU-Kriterium wurde aus der Objekterkennung in die Texterkennung übernommen, zusammen mit dem Schwellwert von 0,5. Wie aber in (Liao et al., 2018) gezeigt wird, ist dieses Kriterium nicht optimal, wenn die detektierte Box für OCR weiterverwendet werden soll.



Abb. 4.8.: OCR Ergebnisse (gelb) für verschiedene erkannte Textboxen mit dem selben IoU. Die rote Box ist die Ground-Truth; die grüne Box das Ergebnis der Texterkennung.

4.1.3.3. Ergebnisse

Für alle EAST-Varianten sind die Ergebnisse im Recall, bei einem IoU Schwellwert von 0,5, fast gleich. 0,91 für das auf den ICDAR 2015 Daten trainierte Modell gegenüber 0,94-0,95 für nur auf den Kartendaten trainierte oder feingetunte Modelle, siehe Abbildung 4.9. Dies zeigt, dass die Erkennung der Texte sich gut von den ICDAR-2015-Daten auf die Altkarten generalisieren lässt und dass das Modell Texte in verschiedensten Situationen gut erkennen kann.

Bei einem IoU Schwellwert von 0,5 sind aber deutliche Unterschied bei der Precision zu beobachten. So erreicht das nur mit den ICDAR 2015 Daten trainierte Modell nur eine Precision von 0,67 verglichen mit Werten von 0,93 bis 0,95 wenn auch

4. Erkennung der Beschriftung von Ortssignaturen

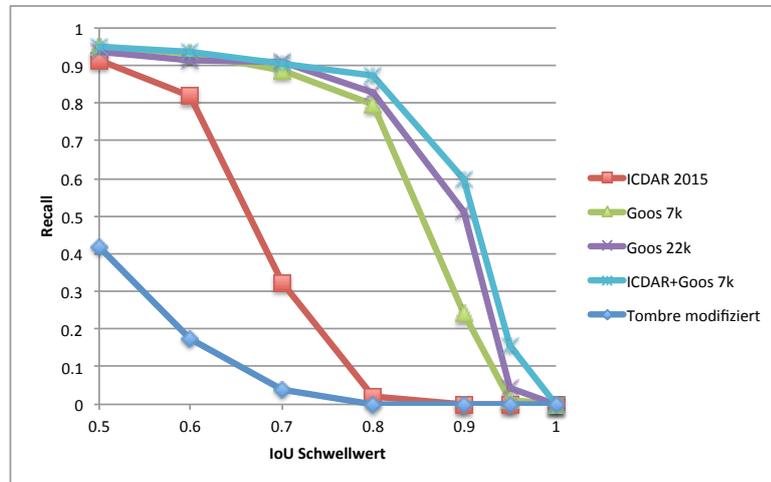


Abb. 4.9.: Recall von EAST und dem modifiziertem Verfahren von Tombre.

Daten von den Altkarten für das Training mitverwendet werden. Eine Betrachtung der fälschlicherweise als Text erkannten Bereiche zeigt, dass diese meist aus regelmäßigen Mustern wie gestrichelten oder schraffierten Bereichen besteht oder Symbole enthalten. Dies sind Muster, die nicht typisch für Fotografien sondern eher für Strichzeichnungen sind und daher nicht in den ICDAR 2015 Trainingsdaten vorkommen. Daher und auch weil bekannt ist, dass regelmäßige Strukturen, die Ähnlichkeiten mit Textstrichen aufweisen, wie Zäune und Gitter, schwerer richtig zu klassifizieren sind (Liu et al., 2018), ist dieses Ergebnis nicht verwunderlich und lässt sich durch geeignete Trainingsdaten korrigieren.

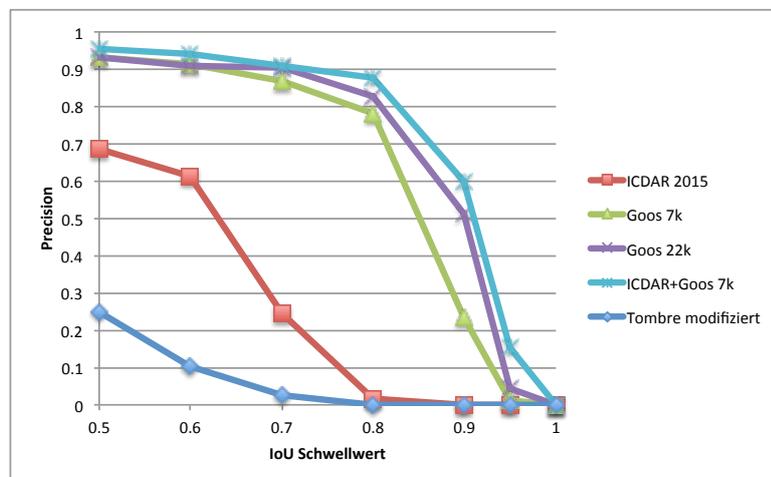


Abb. 4.10.: Precision von EAST und dem modifiziertem Verfahren von Tombre.

4.1. Erkennung von Textbereichen

Der deutlicher Einbruch bei höheren IoU Schwellenwerten liegt vermutlich an der Definition der Bounding-Boxes, einmal mit spezifizierter Ausrichtung des Textes und einmal ohne. Dadurch gibt es immer noch eine Übereinstimmung bei niedrigem Schwellwert. Die bessere Performance bei höheren Schwellwerten der mit der spezifizierten Ausrichtung trainierten oder feingetunten Modelle zeigt aber, dass auch die Bestimmung der ausgerichteten Bounding-Boxes vom Modell gelernt werden kann.

Die Ergebnisse des modifizierten Verfahrens von Tombre liegen in allen Tests deutlich unter den auf Deep-Learning basierenden Verfahren, siehe Abbildungen 4.9, 4.10 und 4.11. Beispiele für die Texterkennung durch beide Ansätze sind in den Abbildungen 4.12 und 4.13 zu sehen.

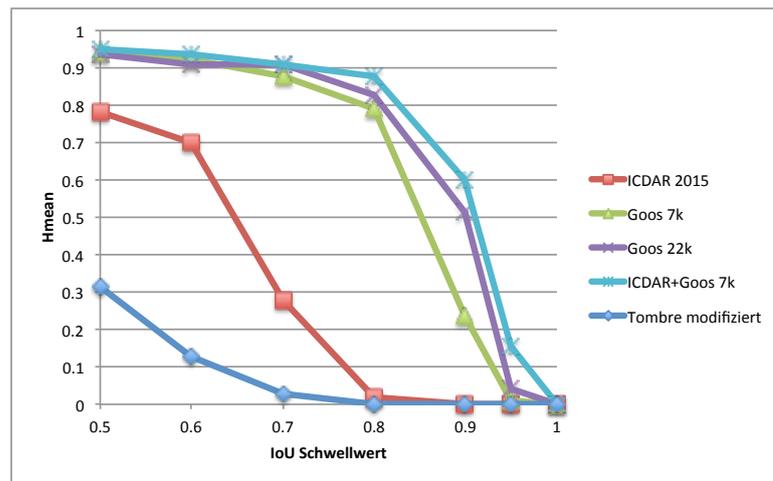


Abb. 4.11.: Hmean von EAST und dem modifiziertem Verfahren von Tombre.

4. Erkennung der Beschriftung von Ortssignaturen



Abb. 4.12.: Erkennung von Ortlabels mittels EAST und Feintuning auf einem Teilbereich der Karte.

4.1. Erkennung von Textbereichen



Abb. 4.13.: Erkennung von Ortslabels durch das an Altkarten angepasste klassische Verfahren zur Text und Grafiktrennung.

4.2. Texterkennung

Dieser Abschnitt behandelt die Transkription von Ortsnamen in Altkarten, nachdem die Position der Ortsnamen schon bestimmt wurde.

Wie bereits im Abschnitt 2.2.2 diskutiert, hat die Texterkennung in Karten mit einigen zusätzlichen Problemen im Vergleich zur Texterkennung in Büchern zu kämpfen:

1. In Büchern ist der Text blockweise angeordnet und kann dann für die Texterkennung in die einzelnen Zeilen zerlegt werden, welche dann normalerweise keine Fremdelemente enthalten. Die Ortsbeschriftungen in Karten hingegen enthalten in ihrer nächsten Umgebung andere Elemente (Ortssignaturen, Kartensymbole, Straßen, Flüsse), welche teilweise auch mitten im Wort auftreten. Dies macht es unmöglich, ohne schon vorher den Text zu erkennen, den Text ohne störende Elemente zu extrahieren.
2. Wie bei der Label-Identifikation erschwert die nicht immer gerade Grundlinie der Texte die OCR-Phase.

Um die zweite Forschungsfrage zu beantworten, wurden zwei Methoden getestet:

1. Ocropy, eine Open-Source OCR-Software zu trainieren;
2. Eine eigene auf CNNs und LSTM aufbauende Architektur zu verwenden.

4.2.1. Ocropy

Ocropy (Breuel et al., 2013) ist eine Open-Source OCR-Engine des OCRopus-Projektes, die für die Experimente in dieser Arbeit gewählt wurde. Ocropy ist eine Sammlung von Kommandozeilentools, die alle Schritte eines OCR-Workflows abdecken. Die eigentliche Zeichenerkennung basiert auf rückgekoppelten Neuronalen Netzen (RNN) mit einer *long short-term memory* Architektur (LSTM) (Hochreiter

und Schmidhuber, 1997) und einer *Connectionist Temporal Classification* (CTC) Ausgabe- und Kostenfunktion (Graves et al., 2006), welche sich mit Transkriptionen auf Zeilenebene trainieren lässt und keine Zerlegung in Zeichen und deren zeichenweise Annotation benötigt. Durch diese Aufteilung in einzelne Tools ist es möglich, jeden der Schritte anzupassen und auch durch eigene Lösungen zu ersetzen. Zuerst wird in Abschnitt 4.2.1.1 der typische Ablauf des Trainings und der anschließenden Erkennung erläutert, wie er für ein Buch durchgeführt wird. Danach werden in Abschnitt 4.2.1.2 die nötigen Änderungen am Workflow erklärt, um Ocropy für das OCR von Kartenlabels verwenden zu können.

Die im Folgenden beschriebenen Schritte beziehen sich auf Version 1.3.3 von Ocropy³, welche am 16.12.2017 veröffentlicht wurde (Ocropy Github, 2018). Der typische Ablauf um ein OCR Modell zu trainieren und anschließend anzuwenden ist folgender.

4.2.1.1. Training und OCR mit Ocropy

Im ersten Schritt müssen die Scans, welche mit ca. 300dpi vorliegen sollen, binarisiert werden. Dies erfolgt mit dem `ocropus-nlbin` Kommando. Als Eingabe werden die Seiten eines Buches erwartet, welche dann gerade ausgerichtet und binarisiert werden. Die Ausgabebilder werden durchnummeriert im Ausgabeordner abgelegt. Die binarisierten Versionen der Eingabebilder erhalten die Endung `.bin.png` und zusätzlich werden unter der Endung `.nrm.png` noch Versionen der Bilder mit normalisierten Graustufen abgelegt. Die Bildversionen mit normalisierten Graustufen sind den binarisierten Bildern sehr ähnlich, sie enthalten aber an den Übergängen zwischen den schwarzen und weißen Bereichen noch Graustufen. Eine Beschreibung des verwendeten Algorithmus gibt es in Afzal et al. (2013).

```
ocropus-nlbin
tests/ersch.png
-o book
```

Auf den binarisierten Scans wird anschließend eine Layoutanalyse durchgeführt, welche eventuell im Bild enthaltene Spalten erkennt und anschließend das Bild in

³<https://github.com/tmbdev/ocropy/releases> (zuletzt abgerufen am 18.06.2018)

4. Erkennung der Beschriftung von Ortssignaturen

seine Zeilen zerlegt. Die einzelnen Zeilen werden in einem Ordner gespeichert. Die in Ocropy enthaltene Layoutanalyse kann mit dem `ocropus-gpageseg` Kommando gestartet werden. Der Parameter `--gray` sorgt dafür, dass zusätzlich zu den binarisierten Zeilen auch die Textzeilen in normalisierten Graustufen ausgegeben werden.

```
ocropus-gpageseg
--gray
'book/?????.bin.png'
```

Die beiden Schritte, Binarisierung und Layoutanalyse, beschreiben die Vorverarbeitung, welche nötig ist, um ein Eingabebild entsprechend vorzubereiten, um danach das Training oder die Texterkennung mit dem OCR Modul durchführen zu können. Für Ocropy werden hier schlussendlich die einzelnen Textzeilen des Dokuments in normalisierten Graustufen oder binarisiert benötigt. Für jede Seite des Dokuments existiert ein nummerierter Ordner, welcher dann die ausgeschnittenen Bilder der einzelnen Zeilen dieser Seite enthält.

Für Ocropy gibt es schon vortrainierte Modelle⁴, um ohne ein eigenes Training eine Texterkennung durchführen zu können. Mit einem speziell für ein Buch angepasstem und trainiertem Modell lassen sich aber besonders für ältere Drucke deutlich bessere Erkennungsraten erreichen. Springmann und Lüdeling (2016) zeigen, dass sich in Büchern, welche von 1487 bis 1870 gedruckt wurden, Zeichenfehlerraten von 1–6% erreichen lassen, wenn ein angepasstes Modell für das jeweilige Buch trainiert wird. Um dieses Training durchzuführen, werden aber Transkriptionen in hoher Qualität benötigt, welche sich nur mit hohem manuellem Aufwand erstellen lassen.

Für das Training bringt Ocropy das `ocropus-rtrain` Tool mit. Es benötigt als Eingabe die Bilder der einzelnen Zeilen in normalisierten Graustufen oder binarisiert und zu jeder Zeile den transkribierten Text, welcher unter dem gleichen Dateinamen aber mit der Erweiterung `.gt.txt` abgelegt sein muss. Die Erstellung dieser Ground-Truth-Dateien wird mit dem `ocropus-gtedit` Tool unterstützt. Mit `ocropus-gtedit` kann für eine Auswahl an Zeilen aus einem Buch eine HTML-Seite erstellen, welche die ausgeschnittenen Bilder dieser Zeilen zusammen mit einem Textfeld für die Transkription anzeigt, wie in Abbildung 4.14 gezeigt. Anschließend

⁴<https://github.com/tmbdev/ocropy/wiki/Models> (zuletzt abgerufen am 18.06.2018)

4.2. Texterkennung

kann das Tool auch genutzt werden um aus der ausgefüllten und abgespeicherten HTML-Seite die transkribierten Texte zu extrahieren und zeilenweise in den für das Training benötigten `.gt.txt` Dateien abzuspeichern.



Abb. 4.14.: Die von ocropus-gtedit erzeugte HTML-Datei zum Erfassen der Ground-Truth-Daten.

```
ocropus-gtedit html
--output correction.html
book/????/01000?.nrm.png
```

4. Erkennung der Beschriftung von Ortssignaturen

```
ocropus-gtedit  
extract  
correction.html
```

Das `ocropus-rtrain` Tool benötigt einen Ausgabepfad und Namen unter dem das trainierte Modell und seine Zwischenversionen abgespeichert werden. Dies wird mit dem `--output` Parameter angegeben. Standardmäßig wird jeweils nach 1.000 trainierten Zeilen eine Zwischenversion abgespeichert und trainiert bis eine Million Zeilen für das Training verwendet wurden. Die in den Ground-Truth-Daten vorhandenen Zeilen werden hierzu mehrfach verwendet, da im Normalfall keine Million unterschiedlicher und annotierter Zeilen für das Training zur Verfügung steht. Zusätzlich wird ein Codec benötigt, welcher alle im Text vorkommenden Zeichen auf eine kontinuierliche Folge von Zahlen, beginnend mit der Null, abbildet. Dies wird benötigt, da intern jedem Zeichen ein Neuron zugeordnet wird, und die einem Zeichen zugeordnete Nummer bestimmt, welches Neuron für die Erkennung des jeweiligen Zeichens zuständig ist. An dieser Stelle sollte auch darauf geachtet werden, dass die Ground-Truth-Daten als NFKC-normalisierte UTF-8-Dateien vorliegen. Ocropy nimmt an, dass die Eingabetexte UTF-8-kodiert vorliegen und führt beim Einlesen automatisch eine NFC-Normalisierung durch, jedoch wird beim Erstellen der Zeichenliste zusätzlich eine NFKC-Normalisierung durchgeführt. Diese zusätzliche NFKC-Normalisierung bildet Zeichen, die die gleiche Bedeutung haben aber verschiedene Darstellungen benutzen, auf ein Zeichen ab. So wird ein langes `s` „f“ auf ein „s“ abgebildet. Da dies nur bei der Erstellung des Codecs passiert, wird das lange `s` und auch alle anderen so normalisierten Zeichen während des Trainings anders normalisiert eingelesen und kann dann keinem Neuron zugeordnet werden. Daher sollten die Trainingstexte normalisiert werden um solche Probleme zu vermeiden.

Schließlich müssen `ocropus-rtrain` noch die Liste der ausgeschnittenen Bilder der Zeilen übergeben werden, die zum Training verwendet werden sollen. Das Training speichert nun nach jeweils 1.000 für das Training verwendeten Zeilen eine Zwischenversion des Modells ab. Hierdurch kann das Training jederzeit unterbrochen werden und eines der abgespeicherten Modelle kann verwendet werden, um eine Texterkennung für beliebige weitere Zeilen durchzuführen.

```
ocropus-rtrain
```

```
--codec 'book/????/??????.gt.txt'  
--output model/ersch  
'book/????/??????.nrm.png'
```

Eine Texterkennung kann nun mittels `ocropus-rpred` durchgeführt werden. Hierzu wird ein trainiertes Modell benötigt: entweder eines der Modelle die im vorherigen Schritt trainiert wurden, oder eines der auf der Ocropy Webseite verfügbaren Modelle⁵. Für die hier im Beispiel verwendete Buchseite ist das Fraktur-Modell geeignet, welches erst noch heruntergeladen und im `models`-Ordner abgelegt werden muss. Schließlich müssen noch die Zeilen, auf denen die Texterkennung durchgeführt werden soll, angegeben werden. Als Ausgabe werden die erkannten Texte zu jeder Zeile in einer gleichnamigen Datei mit der Endung `.txt` abgelegt.

```
ocropus-rpred  
-m models/fraktur.pyrnn.gz  
'book/????/??????.nrm.png'
```

Das `ocropus-hocr` Tool kann nun aus den erkannten Texten eine HTML-Datei im hOCR-Format⁶ generieren. Diese Datei fasst die Texte der einzelnen Zeilen in einer Datei zusammen und speichert Layoutinformationen als zusätzliche Metainformationen ab. Mit Zusatztools ist es auch möglich, diese Datei in andere Formate wie eine durchsuchbare PDF-Datei zu verwandeln.

```
ocropus-hocr  
'book/?????.bin.png'  
-o ersch.html
```

4.2.1.2. Anpassung von Ocropy für OCR von Kartenlabels

Für die Erkennung von Kartenlabels muss der Ablauf angepasst werden. Die Layoutanalyse ist auf die Erkennung von Textblöcken und deren Zerlegung in Zeilen ausgelegt, daher muss dieser Schritt ersetzt werden. Durch den modularen Aufbau von Ocropy als einzelne Kommandozeilenprogramme und die Übergabe

⁵<https://github.com/tmbdev/ocropy/wiki/Models> (zuletzt abgerufen am 18.06.2018)

⁶<http://kba.cloud/hocr-spec/1.2/> (zuletzt abgerufen am 18.06.2018)

4. Erkennung der Beschriftung von Ortssignaturen

der Daten in den nächsten Schritt als Dateien ist dies ohne Modifikationen des Programmcodes möglich. Die Annotation der Textlabels in den Kartendaten wird manuell ausgeführt, was eine hohe Qualität sicherstellt. Die Annotation erfolgt auch nicht als einfache Bounding-Box, welche den Text umschließt, sondern die Grenzen der Bounding-Box sind genauer definiert. Die untere Kante der Bounding-Box liegt auf der Grundlinie und die obere Kante auf der x-Höhe. Mithilfe dieser Daten können dann die einzelnen Labels mit einem gewissen Puffer ausgeschnitten werden, so dass auch alle Ober- und Unterlängen in dem ausgeschnittenen Bildteil enthalten sind.

Die ausgeschnittenen Ortslabels werden dann mit `ocropus-nlbin` binarisiert, wobei beachtet werden muss, dass `ocropus-nlbin` testet, ob die Breite und Höhe des Bildes zwischen 600 und 10000 Pixeln liegt, und somit plausibel für den Scan einer Seite bei 300 dpi erscheint. Dieser Test muss deaktiviert werden, wenn die Binarisierung auf einzelne Zeilen oder Wörter angewendet werden soll. Dies lässt sich über den Schalter `--nocheck` erreichen. Zusätzlich sollte die Korrektur für schief eingescannte Seiten deaktiviert werden, da die Zeilen durch die manuelle Annotation schon richtig ausgerichtet sind. Durch den Parameter `--maxskew 0` kann dies erreicht werden. Als Ergebnis haben wir nun die Ortsnamen jeweils binarisiert und in normalisierten Graustufen in einzelnen Dateien vorliegen.

Um ein Modell trainieren zu können, werden noch die zu den Bildern gehörenden Ground-Truth-Dateien `.gt.txt` benötigt. Wenn die Transkriptionen schon während der manuellen Annotation erfasst wurden, müssen sie nun nur noch in die entsprechenden Dateien exportiert werden. Ansonsten kann wie im Abschnitt 4.2.1.1 das `ocropus-gtedit` Tool verwendet werden, um die Transkriptionen gesammelt im Browser zu erfassen und anschließend in die Ground-Truth-Dateien zu exportieren.

Während des Trainings und der Texterkennung versucht Ocropy die Textzeile zu normalisieren und Zeilenkrümmungen auszugleichen. Da die Textlabels manuell markiert wurden, ist dies unnötig. Durch in die Zeile hineinragende Elemente, wie z.B. Flüsse, Wälder und Orte, wird diese Heuristik gestört, und die Zeilen würden inkonsistent normalisiert. Abbildung 4.15a zeigt Beispiele für Ortslabels, welche auf Höhe der Ober- und Unterlängen aus der Karte ausgeschnitten wurden, Abbildung 4.15b zeigt die Ergebnisse der Zeilennormalisierung. In Abbildung 4.16

ist das selbe für Ortslabels, die mit einem größeren Rand ausgeschnitten wurden, gezeigt. In den normalisierten Versionen, Abbildungen 4.15b und 4.16b, kann man sehen, dass die Ausrichtung und Größe stark schwankt.

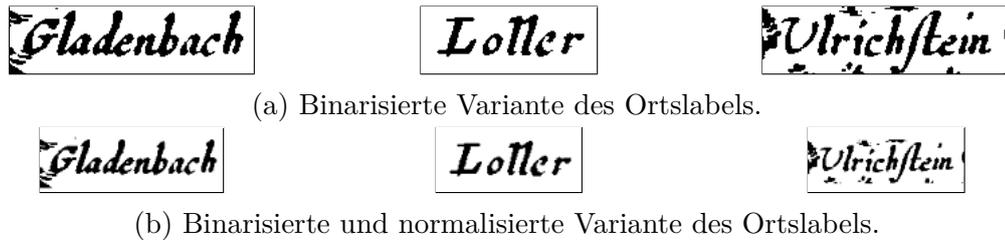


Abb. 4.15.: Beispiele für Ortslabels, die auf Höhe der Ober- und Unterlängen ausgeschnitten wurden.

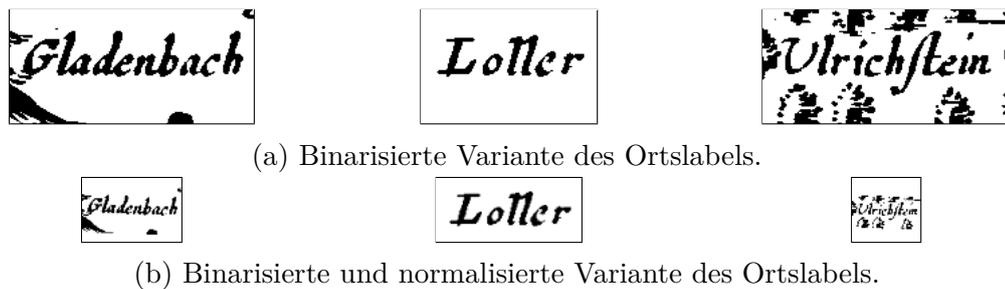


Abb. 4.16.: Beispiele für Ortslabels, die mit einem zusätzlichen Rand ausgeschnitten wurden.

In Abbildung 4.17 wird die Zeichenfehlerrate auf der Testmenge in Abhängigkeit von der Anzahl der Trainingsschritte dargestellt. Das Training erfolgte auf 240 ausgeschnittenen Ortslabels. Einmal wurden die Ortslabels direkt auf Höhe der Ober- und Unterlängen beschnitten (rot dargestellt), und einmal wurde ein zusätzlicher Rand gelassen (blau dargestellt). Beide Varianten wurden mit und ohne die in Ocropy integrierten Zeilennormalisierung trainiert, die Variante mit Zeilennormalisierung wird durch eine gepunktete Linie dargestellt. Nach 20.000 Trainingsschritten gibt es in allen Trainingsvarianten nur noch geringe Veränderungen der Zeichenfehlerrate. Die Zeichenfehlerraten der beiden Beschnittvarianten ohne Verwendung der Zeilennormalisierung bewegen sich im gleichen Bereich. Die Varianten mit Zeilennormalisierung zeigen dagegen eine deutlich höhere Fehlerrate. Die Beeinträchtigung durch die inkonsistente Normalisierung geht so weit, dass

4. Erkennung der Beschriftung von Ortssignaturen

das Modell die Erkennung von Zeichen nicht lernen kann, wenn ein zusätzlicher Rand um die Ortslabels vorhanden ist.

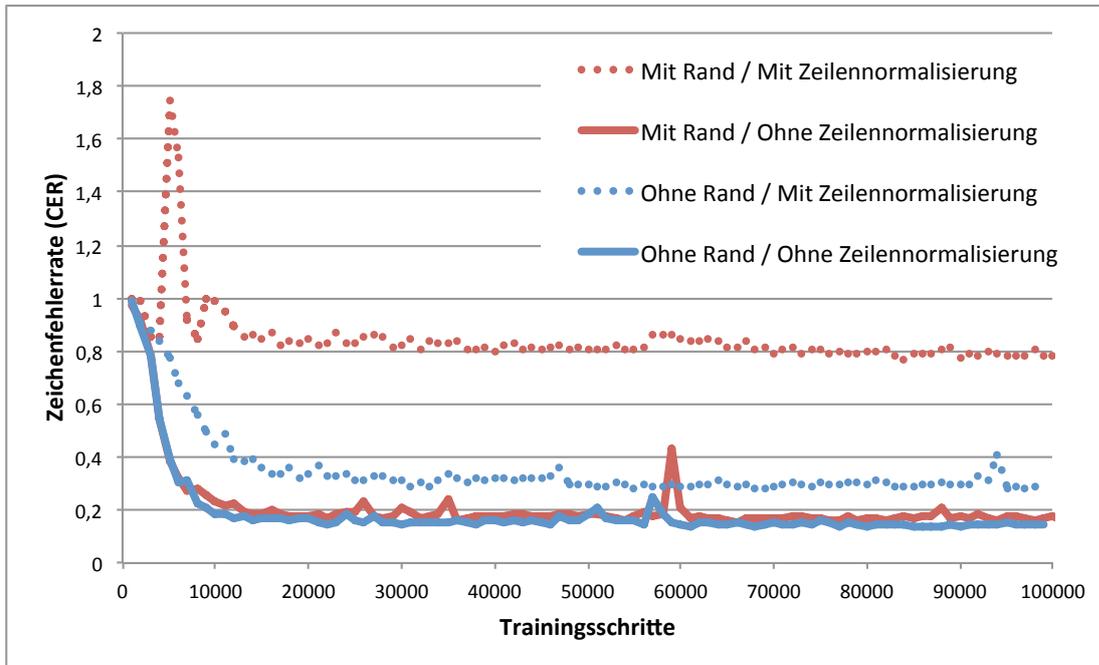


Abb. 4.17.: Vergleich der Zeichenfehlerrate auf einem Testset, wenn Ocropy mit oder ohne Zeilennormalisierung trainiert wird.

Um Ocropy dennoch zur Texterkennung der Ortslabels verwenden zu können, musste dieser Vorverarbeitungsschritt zunächst umgangen werden. Die Zeilennormalisierung ist in `ocropus-rtrain` und `ocropus-rpred` eingebaut und wird standardmäßig angewendet. Beide Programme unterstützen den Schalter `--nolineest` um die interne Zeilennormalisierung abzuschalten, es werden dann aber als Eingabe Dateien mit der Endung `.dew.png` erwartet und nicht mehr die binarisierten `.bin.png` oder normalisierten Graustufen `.nrm.png` Dateien akzeptiert. Die normalisierte Variante `.dew.png` einer Zeile lässt sich sowohl mit `ocropus-dewarp` erzeugen als auch mit `ocropus-rtrain --dewarp`. Da in diesem Fall keine Normalisierung der Zeile durchgeführt werden soll, kann die binarisierte Version umkopiert werden mit der Dateiondung `.dew.png`. Beim umkopieren muss darauf geachtet werden, dass alle Zeilen die gleiche Höhe aufweisen, welche auch mit der vom Modell erwarteten Eingabehöhe, die standardmäßig bei 48 Pixeln liegt, übereinstimmen muss. Die vom Modell erwartete Eingabehöhe lässt sich auch über den `--height`

Parameter von `ocropus-rtrain` anpassen. Der Ersatz für die Zeilennormalisierung besteht nun aus einer Umskalierung, welche das Seitenverhältnis beibehält, der binarisierten Zeile auf die vom Modell erwartete Eingabehöhe und die Änderung der Dateiendung auf `.dew.png`.

4.2.2. Training eines eigenen Modells

Teile dieses Abschnitts wurden auf dem auf dem Exploring Old Maps Workshop 2017 vorgestellt. (Höhn, 2017)

Die hier vorgestellte Zeichenerkennung für Ortslabels ist ähnlich aufgebaut wie die in `Ocropy`⁷ verwendete Zeichenerkennung. Es wird ein bidirektionales rückgekoppeltes neuronales Netzwerk, speziell ein LSTM (Hochreiter und Schmidhuber, 1997), verwendet. Das LSTM erzeugt für jeden Zeitschritt der Eingabe, was einer Spalte der Eingabematrix entspricht, eine Ausgabe. Eine Connectionist Temporal Classification (CTC) Ausgabe- und Verlustfunktion (Graves et al., 2006) wird verwendet um die Vorhersage der Zeichensequenz zu lernen. Im Gegensatz zu `Ocropy` wird das Eingabebild nicht direkt als Eingabe für das LSTM verwendet. Stattdessen wird eine Reihe von abwechselnden CNN- und Max-Pool-Schichten verwendet um Merkmalsvektoren aus dem Eingabebild zu erzeugen. Die Ausgabe der CNN Schichten wird dann als Eingabe für das LSTM verwendet.

Rückgekoppelte neuronale Netze verarbeiten eine Eingabesequenz schrittweise. In jedem Schritt wird ein Element der Eingabesequenz als Eingabe des RNNs benutzt und eine Ausgabe berechnet. Zusätzlich wird noch ein interner Zustandsvektor berechnet, welcher als eine zusätzliche Eingabe für den nächsten Schritt dient. Diesen versteckten Zustand kann man sich als eine Art Kurzzeitgedächtnis vorstellen, in dem Informationen aus vorherigen Schritten gespeichert werden und in die Berechnung der späteren Schritte eingehen können. Schon von Hochreiter (1991) wurde gezeigt, dass der Fehler beim Zurückpropagieren durch die Zeit nach nur wenigen Schritten zu klein oder zu groß wird (bekannt als *vanishing/exploding gradient problem*) und ein effektives Lernen verhindert. LSTM (Hochreiter und Schmidhuber, 1997), was im Deutschen langes Kurzzeitgedächtnis bedeutet, ist

⁷<https://github.com/tmbdev/ocropy> (zuletzt abgerufen am 18.06.2018)

4. Erkennung der Beschriftung von Ortssignaturen

eine Variante von rückgekoppelten neuronalen Netzen (RNNs), die dafür ausgelegt ist, diese Probleme zu umgehen und auch Abhängigkeiten über lange Zeiträume, mehrere tausend Schritte, lernen zu können. Erreicht wird dies durch die Einführung von Gates, welche das Vergessen, das Hinzufügen neuer Informationen und die Ausgabe kontrollieren, ansonsten aber ein konstantes Zurückpropagieren von Fehlern über lange Zeiträume ermöglichen.

Ähnliche Architekturen wurden schon für verschiedene Texterkennungsaufgaben eingesetzt. Wie schon beschrieben arbeitet Ocropy mit einem bidirektionalen RNN für die Texterkennung. Auch Tesseract ab Version 4.0 unterstützt dies und enthält sogar eine eigene Konfigurationssprache (Variable-size Graph Specification Language (VGSL))⁸, um die Architektur anzupassen. Es lassen sich verschiedenste Kombinationen aus CNNs und RNNs realisieren und mit der CTC Ausgabe- und Kostenfunktion verbinden. Besonders im Bereich der Scene Text Recognition werden Kombinationen aus CNNs und RNNs eingesetzt (Shi et al., 2016). Für die Texterkennung in Altkarten bestehen aber einige Unterschiede zu den hier genannten Arbeiten, in erster Linie die nur sehr geringe Menge an verfügbaren Trainingsdaten. Daher liegt der Fokus dieser Arbeit auf dem Finden einer Architektur, welche sich mit möglichst wenig Trainingsdaten trainieren lässt.

Das Netzwerk besteht aus zwei Convolutional-Layers, die jeweils von einem Max-Pool-Layer gefolgt werden, einem bidirektionalem 1D-LSTM, einer Dropout- und CTC-Schicht, siehe Abbildung 4.18. Die Kernelbreite und Kernelhöhe kann aus der Menge $\{1, 3, 5, 7\}$, die Anzahl der Filter aus $\{2, 4, 8, 16, 32, 64, 128\}$ frei gewählt werden. Für das LSTM kann die Anzahl der internen Zustände aus $\{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$ und die Dropout-Rate von 0% – 100% gewählt werden. Innerhalb dieser Grenzen sollte das Netzwerk gefunden werden, welches mit wenigen Trainingsdaten möglichst effizient lernt. Hierzu wurde eine zufällige Suche (engl.: *random rearch*) verwendet, hierbei werden zufällige Parameterkombinationen getestet. Im Gegensatz zu einer Rastersuche, bei der alle Parameter systematisch in bestimmten Abständen durchprobiert werden, ist die zufällige Suche effizienter (Bergstra und Bengio, 2012). Als Optimierungskriterium wurde genutzt, wie schnell das Modell für eine computergerenderte Texte eine fehlerfreie Erkennung lernen kann. Das in Abbildung 4.18 gezeigte Modell wurde durch diese Suche gefunden.

⁸<https://github.com/tesseract-ocr/tesseract/wiki/VGSLSpecs> (zuletzt abgerufen am 18.06.2018)

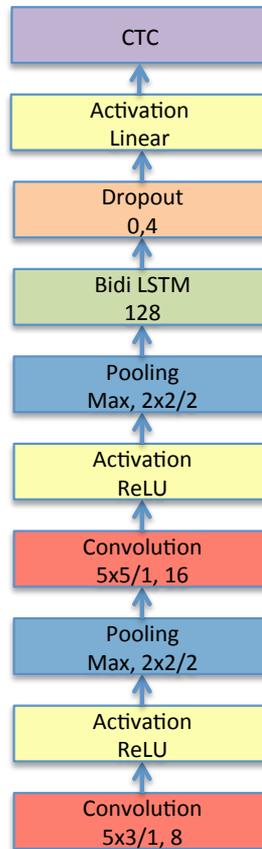


Abb. 4.18.: Struktur des Netzwerks

Um den Effekt des Overfitting zu reduzieren, vergrößern wir künstlich die Menge der zur Verfügung stehenden Trainingsdaten wie in (Simard et al., 2003) beschrieben. Den Bildern wird ein gaußsches Rauschen hinzugefügt, und die Bilder werden zufällig verzerrt indem jedes Pixel um einen Verschiebungsvektor verschoben wird. Die Verschiebungsvektoren werden durch das gaußsche Glätten eines normalverteilten Rauschens bestimmt. Beispiele für diese Bildmodifikationen werden in Abbildung 4.19 gezeigt.

Eine weitere Möglichkeit zur Reduktion von Overfitting ist Dropout. Durch Dropout wird während des Trainings ein festgelegter Anteil, z.B. 30%, an Neuronen und ihre Verbindungen zufällig entfernt, wobei die Aktivitäten der verbleibenden Neuronen entsprechend skaliert werden, um den Erwartungswert der Aktivitäten konstant zu halten. Eine mögliche Interpretation des Dropouts besteht darin, dass während des Trainings verschiedene ausgedünnte Versionen des Netzwerks trainiert werden.

4. Erkennung der Beschriftung von Ortssignaturen

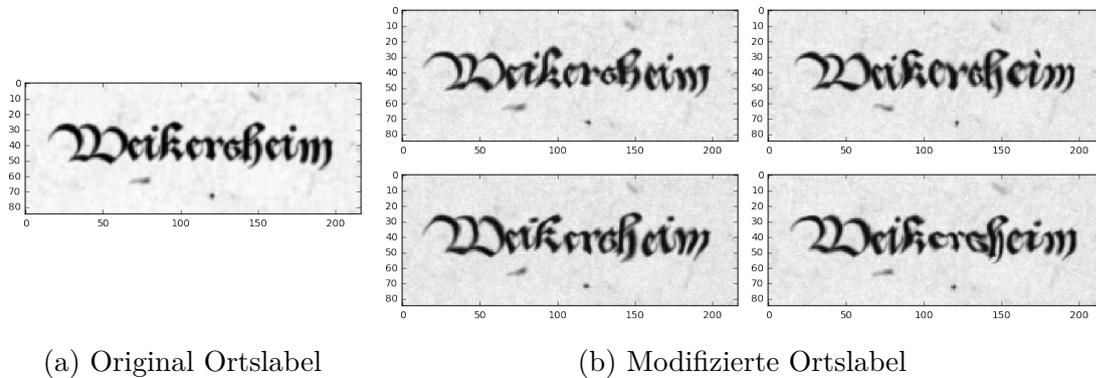


Abb. 4.19.: Beispiele für modifizierte Bilder.

Während der Testphase wird dann das Mitteln der Vorhersagen dieser ausgedünnten Netze simuliert, indem die Vorhersage des vollen Netzwerkes mit entsprechend reduzierten Gewichten verwendet wird. (Srivastava et al., 2014)

Um die Geschwindigkeit des Trainings und die Erkennungsgenauigkeit weiter zu erhöhen, wird ein schon vortrainiertes Modell verwendet und mittels Feintuning an die jeweilige Karte angepasst. Das Vortraining wird mit generierten Textlabels durchgeführt. Hierzu werden zufällig gewählte Ortsnamen⁹ in 110 verschiedenen Frakturschriften aus der Schalansky (2008) beiliegenden CD-ROM auf ein leeres Stück Kartenhintergrund gerendert. Die Schriften werden in verschiedenen Graustufen, mit 10% Transparenz gerendert und mit verschiedenen starken gaußschen Weichzeichnern unscharf dargestellt. Diese mit Transparenz gerenderte Schrift wird nun mit dem Hintergrund kombiniert. Abbildung 4.20 zeigt einige so generierte Textlabels.



Abb. 4.20.: Beispiele für generierte Textlabels

⁹<http://www.ling.uni-potsdam.de/~kolb/DE-Ortsnamen.txt> (zuletzt abgerufen am 19.06.2018)

Das OCR-System ist in Python implementiert und verwendet TensorFlow als Deep-Learning-Framework. TensorFlow ist eine Programm-bibliothek für maschinelles Lernen, wird aktiv von Google entwickelt und ist als Open-Source-Software verfügbar. TensorFlow ist aufgeteilt in ein Backend, welches hauptsächlich in C++ und CUDA implementiert ist und die in Form eines Graphen definierten Berechnungen ausführt. Der Graph, welcher die Berechnung definiert, lässt sich mithilfe des in Python implementierten Frontends definieren. Mittlerweile gibt es auch Frontends in Java und Go, wobei diese noch nicht alle Operationen unterstützen und es empfohlen wird, diese Sprachen zum Laden und Ausführen von in Python definierten Modellen zu nutzen. (Abadi et al., 2016; TensorFlow Java, 2018; TensorFlow Go, 2018)

Für das Training werden die Labels unter Beibehaltung des Seitenverhältnisses auf eine Höhe von 85 Pixeln skaliert. Dies ist nötig, da das 1D-LSTM zwar mit verschiedenen langen Sequenzen arbeiten kann, aber die Dimension des Merkmalsvektors für ein Modell fest ist. Bei der Umskalierung wird die x-Höhe auf 17 Pixel normalisiert und ein Bereich von 34 Pixeln über und unter, sowie von 17 Pixeln links und rechts der Annotation mit ausgeschnitten. Ein ausgeschnittenes Ortslabel mit roten Linien auf Höhe von 34 Pixeln und 51 Pixeln, welche der erwarteten Position von Grundlinie und x-Höhe entsprechen, ist in Abbildung 4.21 dargestellt.



Abb. 4.21.: Ortslabel mit Überlagerten roten Linien auf Höhe von 34 Pixeln und 51 Pixeln.

4.2.3. Evaluierung

Die zwei Ansätze, Ocropy und die eigene Implementierung eines OCR, werden mit verschiedenen großen Trainingsmengen an ausgeschnittenen Ortslabels und den dazugehörigen transkribierten Texten trainiert, um die erreichte Fehlerrate auf einem Testset zu vergleichen. Zusätzlich wird auch der Einfluss eines Vortrainings

4. Erkennung der Beschriftung von Ortssignaturen

untersucht, d. h. das Modell wird zuerst auf gerenderten Texten trainiert und anschließend wird ein Feintuning mit den eigentlichen Trainingsdaten durchgeführt.

Die Ansätze wurden auf einer Altkarte (A.2.4) mit 929 Ortslabels ausgewertet von denen 260 für das Testset verwendet wurden und der Rest für das Training zur Verfügung stand. Für alle verwendeten Ortslabels waren zeichengenaue Transkriptionen vorhanden.

Es wird getestet, welche Zeichenfehlerraten sich in Abhängigkeit von der Anzahl der verwendeten Trainingslabels erreichen lassen und wie das Vortraining die Fehlerrate beeinflusst.

Zur Evaluierung wurde die Zeichenfehlerrate (CER) gewählt. Die Zeichenfehlerrate wird pro Label berechnet und anschließend der Mittelwert gebildet um die Zeichenfehlerrate für das Testset zu erhalten.

$$\text{CER}_{\text{Label}} = \frac{\text{EditDistanz}(\text{KorreakterText}, \text{ErkannterText})}{\text{Zeichenanzahl}(\text{KorreakterText})} \quad (4.4)$$

Mit deaktivierter Normalisierung der Grundlinie war es möglich Ocropy zu trainieren, und es ließ sich eine Zeichenfehlerhäufigkeit (character error rate, CER) von 19,6% erreichen, wenn 200 Labels für das Training verwendet wurden. Das hier vorgestellte OCR-System erreicht auf den gleichen Trainingsdaten ohne Vortraining eine Zeichenfehlerrate von 13,4%. Diese reduziert sich weiter auf 4,8%, wenn das vortrainierte Modell mit den Trainingsdaten feingetunt wird. Das nur auf künstlichen Daten vortrainierte Modell ist nicht ausreichend um selbst eine Erkennung auf den realen Daten zu ermöglichen und weist hier eine CER von ca. 90% auf. Einige ausgewählte Beispiele für Ortslabels und den jeweils erkannten Text sind in Abbildung 4.22 abgebildet.

Für das Vortraining wurden zwei Varianten untersucht, es wurden einmal moderne Ortsnamen genutzt und in der zweiten Variante wurden zufällige Zeichenketten genutzt um diese zu rendern und das Vortraining damit durchzuführen. Abbildung 4.23 zeigt die erreichten Fehlerraten in Abhängigkeit von der Anzahl an Trainingslabels. Es werden die Ergebnisse von Ocropy mit dem in dieser Arbeit vorgestellten

4.2. Texterkennung

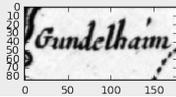
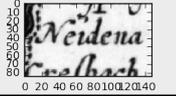
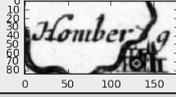
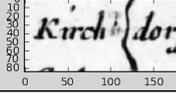
	Gundelheim
	Weispurg
	Neidena
	Erefbach
	Homberg
	Kirchdorff

Abb. 4.22.: Ortslabels mit erkanntem Text, Fehler sind rot markiert. Das Modell wurde mit Vortraining und Feintuning auf 200 Labels über 400 Epochen trainiert. Die Zeichenfehlerrate des Modells auf dem Testset beträgt 5.3%.

OCR-Modell, ohne Vortraining und mit den zwei verschiedenen Varianten für das Vortraining, verglichen.

Das auf Ortsnamen vortrainierten Modell schneidet besser ab, wenn nur sehr wenige Trainingsdaten vorhanden sind, hier für unter 100 Trainingslabels. Für größere Trainingsmengen schneidet das auf zufälligen Zeichenketten trainierte Modell zwischen 0,5% und 1,5% besser ab, siehe Abbildung 4.24. Eine mögliche Erklärung hierzu ist, dass das auf Ortsnamen trainierte Modell ein implizites Sprachmodell lernt, wobei dieses Sprachmodell auf modernen Ortsnamen beruht und damit nicht exakt zu den historischen Schreibweisen in der Altkarte passt. Für sehr geringe Trainingsmengen könnte dies aber dennoch einen Vorteil durch dieses implizite Sprachmodell entstehen und bei größeren Trainingsmengen sich dann nachteilig auswirken. In (Sabir et al., 2017) wird die Auswirkung des von LSTMs gelernten impliziten Sprachmodells im OCR-Kontext untersucht. Diese Arbeit zeigt eine Verbesserung der Fehlerrate von 2.4% durch das implizit gelernte Sprachmodell auf synthetischen Daten gegenüber der Erkennung von zufälligen Sequenzen. Dies unterstützt die Vermutung, dass ein implizit gelerntes Sprachmodell für diese

4. Erkennung der Beschriftung von Ortssignaturen

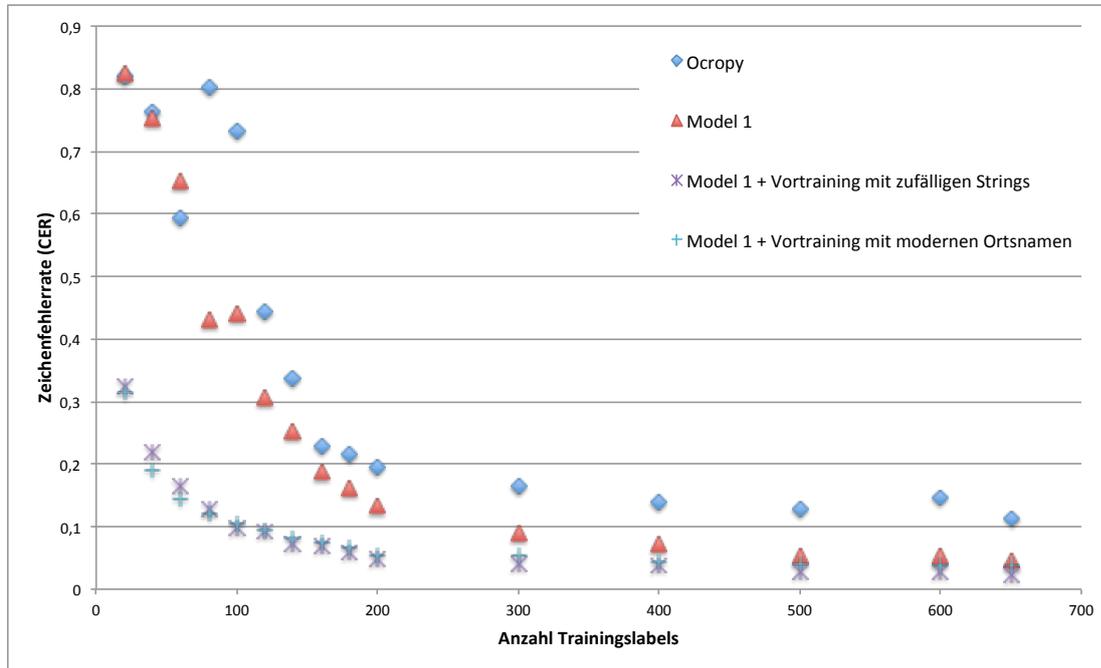


Abb. 4.23.: Vergleich der Zeichenfehlerraten von Ocropy und dem eigenen OCR-Modell.

Unterschiede verantwortlich ist. Eine Frage, die bleibt, ist, ob das LSTM auch für die zufälligen Trainingssequenzen ein implizites Sprachmodell lernt und inwieweit dieses während des Feintunings angepasst wird. Für das implizite Sprachmodell der Ortsdaten findet ja anscheinend keine vollständige Anpassung statt.

4.3. Diskussion der Ergebnisse

Für die Erkennung von Textbereichen ist die angepasste klassische Methode aus Abschnitt 4.1.1 verglichen mit der nur auf Fotos trainierten Version des Deep-Learning-basierten EAST (Zhou et al., 2017) mindestens um den Faktor zwei schlechter für die Metriken Precision und Recall. Die auf den Kartendaten feinetunten Versionen von EAST zeigen eine nochmals deutlich höhere Performanz, besonders für höhere IoU Schwellwerte. Somit können die feinetunten Varianten Annotationen für Textbereiche liefern die deutlich näher an den an der Grundlinie und x-Höhe ausgerichteten manuellen Annotationen liegen. Dies führt zu einer konsistenteren Ausrichtung der Texte in den ausgeschnittenen Labels und sollte ein

4.3. Diskussion der Ergebnisse

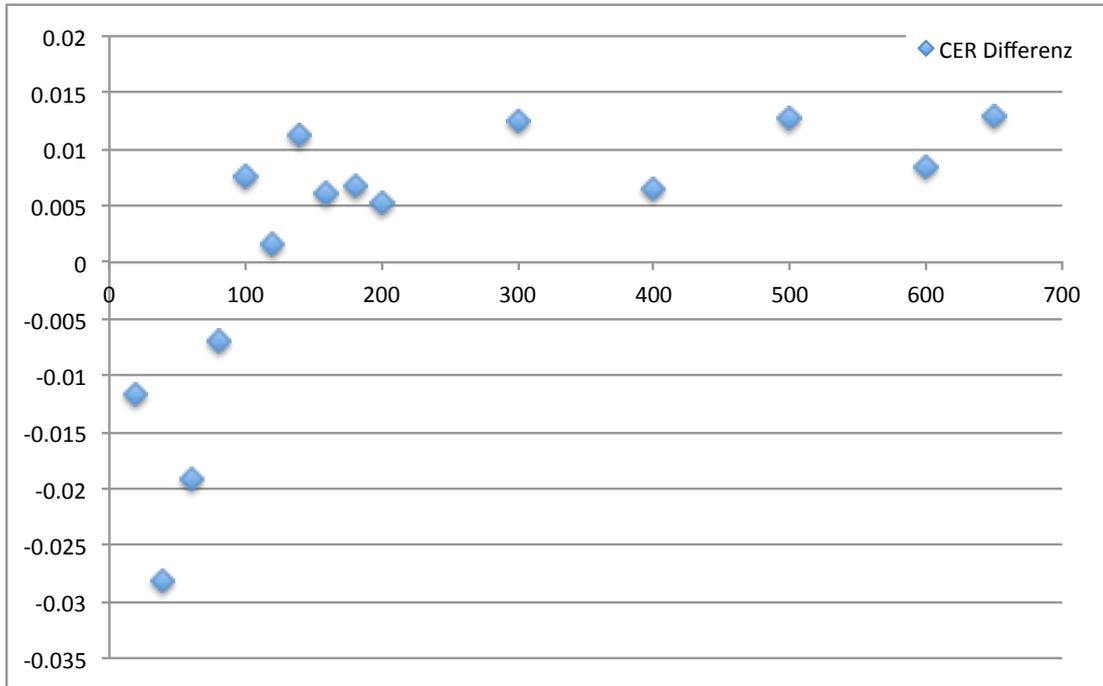


Abb. 4.24.: Differenz der Zeichenfehlerrate von Vortraining mit gerenderten Ortsnamen und zufälligen Zeichenketten in Abhängigkeit von der Anzahl der Trainingslabels.

OCR erleichtern. Das Zusammenspiel zwischen der Erkennung von Textbereichen und den hier vorgestellten OCR-Ansätzen wurde aber nicht untersucht, die Performanz der OCR-Ansätze wurde auf den manuell annotierten Daten bestimmt.

Da CNNs in der Bilderkennung im Allgemeinen zur deutlichen Verbesserung der Erkennungsrate beitragen, und Ocropy nur aus einem LSTM-Layer besteht, wird für diese Arbeit eine Kombination aus Convolutional- und LSTM-Layern verwendet, welche gezielt für ein schnelles Lernen ausgewählt wurde. Die Reduzierung der Fehlerrate konnte durch eine Kombination aus einer Vergrößerung des zur Erkennung genutzten Bildbereiches sowie Vortraining auf generierten Daten, Data-Augmentation und der optimierten Architektur erreicht werden. Somit wurde eine Architektur geschaffen, die gezielt dem schnellen Lernen mit geringen Datenmengen dient. Durch die Kombination dieser Techniken ist es möglich, eine CER von ca. 5% nach dem Training mit nur 200 Labels (Ortsnamen) zu erreichen. Das entspricht gegenüber Ocropy einer Fehlerreduktion um den Faktor vier.

4. Erkennung der Beschriftung von Ortssignaturen

Folgende Erkenntnisse ließen sich nach den Experimenten feststellen:

- Ein Training kann mit leicht erhöhter Fehlerrate auf Karten mit ähnlicher Schrift übertragen werden.
- Braucht präzise ausgerichtete Texte, stellt hohe Anforderungen an die Annotation.
- Training auf künstlichen Daten war nicht übertragbar auf die realen Daten.

4.4. Zusammenfassung

Dieses Kapitel hat die möglichen Lösungen vorgestellt, um die zweite Forschungsfrage zu beantworten:

Wie lässt sich die Beschriftung der Ortssignaturen automatisiert erfassen im Sinne von Label-Identifikation und Label-Texterkennung?

Dazu wurde der Prozess der Erkennung in zwei Teilschritte zerlegt: Erkennung von Textbereichen und Texterkennung in diesen Bereichen.

Zu jedem der Teilschritte, Erkennung von Textbereichen und OCR, wurden jeweils zwei Lösungsansätze vorgestellt. Insgesamt wurden die folgenden Techniken analysiert:

1. Erkennung der Textbereiche:
 - a) Regelbasierte Verfahren zur Klassifizierung von Zusammenhangskomponenten
 - b) Convolutional Neural Networks
2. Texterkennung:
 - a) Anpassung und Training von Ocropy
 - b) Neural Networks Eigenlösung

Die Experimente haben gezeigt, dass im Bereich der Erkennung von Textbereichen

4.4. Zusammenfassung

existierende Deep-Learning-basierten Verfahren durch ein Training mit geeigneten Daten eine Erkennung mit sehr hoher Genauigkeit erreichen können. Für die Texterkennung konnte durch ein eigenes OCR die Fehlerrate im Vergleich zu Ocropy auf ein Viertel gesenkt werden.

5. Verknüpfung mit einer geographischen Datenbank

Dieses Kapitel behandelt die Verknüpfung der einzelnen in einer Karte vorkommenden Ortssignaturen mit einer Ortsdatenbank. In vielen Online-Kartenportalen sind die Karten nur oberflächlich Georeferenziert, d. h. es sind nur wenige markante Punkte mit Geokoordinaten verknüpft, und aus diesen wird für die Karte eine Transformation zwischen Raster- und Geokoordinaten bestimmt. Dieses Vorgehen ist auch in Abschnitt 2.4 für die Georeferenzierung in GIS-Software beschrieben. Dies wird dann genutzt um den von der Karte abgedeckten geographischen Bereich zu bestimmen. Hier steht aber die Erfassung aller auf der Karte dargestellten Orte im Vordergrund. Dadurch werden aber auch jedem verknüpften Ort der Altkarte Geokoordinaten zugeordnet, da diese für jeden Ort in der Ortsdatenbank auch vorliegen.

Die Verknüpfung mit geographischen Koordinaten erfolgt über die Verknüpfung des zu referenzierenden Ortes auf der Altkarte mit einem Ortseintrag in einer Ortsdatenbank. Ein Ortsname kann mehrfach vorkommen, es ist also eine eindeutige Referenzierung, z.B. über einen Schlüssel, notwendig. Die Geokoordinaten eines Ortes sind als Schlüssel nicht geeignet, da diese einen Punkt repräsentieren, ein Ort aber eine Fläche besitzt und verschiedene Geokoordinaten ausgewählt werden können um den Ort zu repräsentieren. Daher wird für die Referenzierung eine einheitliche Ortsdatenbank, welche für alle zu referenzierenden Werke verwendet wird, benötigt.

In den folgenden Unterabschnitten sollen Ansätze und Ergebnisse vorgestellt werden, um eine möglichst effiziente Suche nach dem entsprechenden Ortseintrag aus dem Ortsverzeichnis zu ermöglichen.

5.1. Das Problem der Georeferenzierung von Ortsmarkern in Altkarten

Das Auffinden der Orte sollte trotz abweichender Schreibweise, in den Altkarten, ohne großen Aufwand erfolgen können. Im Deutschen gab es bis Mitte des 19. Jahrhunderts keine einheitliche Rechtschreibung, es gab sogar zwei prominente Lager die grundverschiedene Herangehensweisen an die Rechtschreibung propagierten, zum einen „Schreib, wie du sprichst“ uns zum anderen eine an die Wortherkunft angelehnte Schreibweise (von Raumer, 1863, S. 108ff). Dies führte zu einer großen Bandbreite an Schreibweisen. So konnten im selben Text eines Autors, ja sogar im selben Satz, verschiedene Schreibweisen eines Wortes oder Namens verwendet werden. Zusätzlich haben Orte im Lauf der Zeit ihren Namen und dessen Schreibweise noch aus vielen weiteren Gründen geändert. So wurden den Namen Bestandteile hinzugefügt um zwei Orte zu unterscheiden, z. B. durch die Präfixe Ober-/Unter- und Neu-, oder bestimmte Rechte auszudrücken, z. B. durch das Präfix Markt-. Auch wurden Ortsnamen ins Lateinische übersetzt und zu einer späteren Zeit wieder zurück übersetzt und waren im Allgemeinen auch den Lautänderungen und Verschleifungen unterworfen. Dies führt zu einer großen Bandbreite an Schreibweisen die über die Zeit und auch zu einer Zeit in Verwendung waren als auch verschiedenen Namen die für einen Ort in Verwendung waren.

Durch diese vielfältigen Namensänderungen wird teilweise eine Zuordnung von Ortsnamen ohne eine zusätzliche Recherche in anderen Dokumenten unmöglich sein. Ein weiterer Ansatz der hier nicht in Betracht gezogen wird aber zusätzlich eingesetzt werden kann, ist das Aufbauen und Verwenden einer Datenbank von Ortsnamensvarianten. Hierdurch können die Ergebnisse von Ortsnamensrecherchen in zusätzlichen Karten wiederverwendet werden.

5.2. Heuristiken zur Verlinkung von Ortsmarkern mit einer geographischen Datenbank

Um die Georeferenzierung von Ortsmarkern zu unterstützen wird eine Datenbank mit Orten und den zugehörigen Geokoordinaten verwendet. Um das Finden eines Ortes in der Ortsdatenbank zu unterstützen wird ein Ranking der Orte erstellt. Dieses Ranking basiert auf der Position des Ortes und dem Ortsnamen. Mithilfe der schon georeferenzierten Orte wird eine projektive Transformation zur Georeferenzierung der Gesamtkarte berechnet. Durch diese Transformation lässt sich aus den Pixelkoordinaten einer Ortssignatur deren ungefähre geographische Lage ableiten. Die Karten enthalten Ungenauigkeiten und Verzerrungen wie in Abbildung 5.1 zu sehen ist. Durch eine projektive Abbildung können nur gerade Linien auf gerade Linien abgebildet werden, daher bleiben auch nach der Transformation noch Ungenauigkeiten in der Vorhersage der Geoposition erhalten. Auch mit diesen Ungenauigkeiten kann diese Transformation genutzt werden um den Suchbereich einzuschränken. Zusätzlich kann der in der Karte verwendete Ortsname genutzt werden um diesen mit den modernen Ortsnamen zu vergleichen und ein Ranking der Orte zu erstellen.

Da die in dieser Arbeit verwendeten Karten größtenteils deutschsprachige Regionen abdecken, werden auch auf die deutsche Sprache abgestimmte Verfahren für die Vorschlagsgenerierung eingesetzt, wie phonetische Algorithmen und für die Analyse der Wortbestandteile. Phonetische Algorithmen bilden ein Wort auf einen Code ab, meist eine Buchstaben-Zahlenfolge, der für gleich oder ähnlich klingende Wörter übereinstimmen soll. Die Kölner Phonetik (Postel, 1969) ist ein phonetischer Algorithmus ähnlich zu Soundex oder Metaphone aber für die deutsche Aussprache optimiert. Die Kölner-Phonetik bildet die Buchstaben, abhängig von ihrem Kontext und Position, auf die Zahlen 0 – 8 ab. Die Buchstabencodierungen sind in Tabelle 5.1 aufgelistet. Nach der Codierung werden benachbarte doppelte Ziffern entfernt und dann alle 0-Codes außer an der ersten Position.

Die Kölner-Phonetik wurde in Höhn et al. (2013) mit den folgenden Erweiterungen genutzt um die Vorschlagsliste einzuschränken. Die verwendete Suche ist als Präfixsuche ausgelegt. Präfixe die über die Zeit zu Ortsnamen hinzugefügt wurden werden nacheinander wieder entfernt und alle entstehenden Varianten werden

5. Verknüpfung mit einer geographischen Datenbank

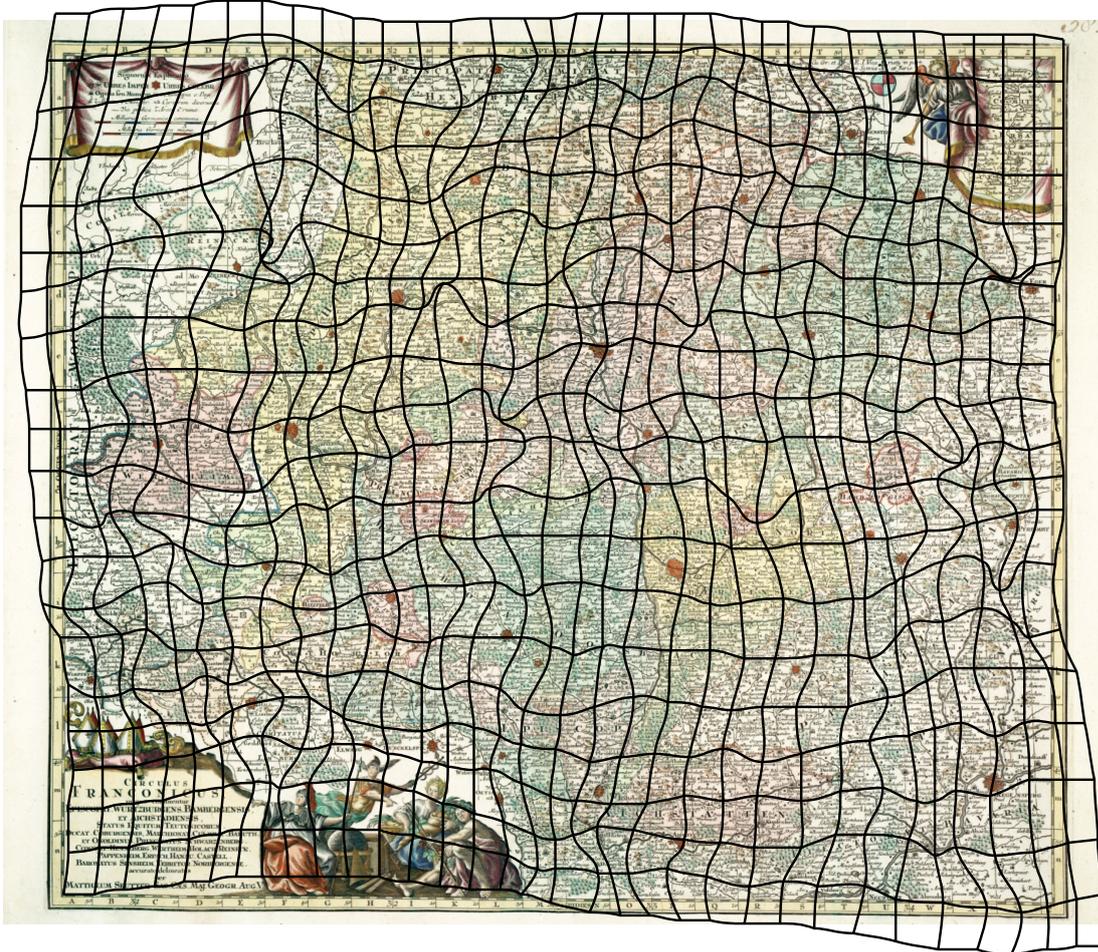


Abb. 5.1.: Verzerrungsgitter mit einem Linienabstand von 0.1° , Berechnet mit Map-Analyst (Jenny und Hurni, 2011) aus ca. 3,200 mit RAT georeferenzierten Orten.

für die phonetische Suche verwendet. Auch werden die modernen Ortsnamen an allen Nicht-Buchstaben-Zeichen aufgetrennt und es wird ein neuer Substring erzeugt. So ergeben sich z. B. für „Bad Rippoldsau-Schapbach“ die Substrings „Bad Rippoldsau-Schapbach“, „Rippoldsau-Schapbach“ und „Schapbach“ welche dann für die phonetische Suche verwendet werden. Dadurch sind die früheren Namen „Rippoldsau“ und „Schapbach“ Präfixe oder Übereinstimmungen mit einem der generierten Substrings. Zusätzlich werden auch typische Präfixe welche zu Ortsnamen hinzugefügt wurden wieder entfernt, wie z. B. Markt-, Burg-, Münch- und Mönch-. Ähnliche sprachspezifische Ressourcen können auch für Karten verwendet werden welche andere Regionen und damit andere Sprachen abdecken.

5.2. Heuristiken zur Verlinkung von Ortsmarkern mit einer geographischen Datenbank

Tab. 5.1.: Codierungstabelle der Kölner-Phonetik (Postel, 1969)

Buchstabe	Kontext	Code
A, E, I, J, O, U, Y		0
H		-
B		1
P	nicht vor H	1
D, T	nicht vor C, S, Z	2
F, V, W		3
P	vor H	3
G, K, Q		4
C	im Anlaut vor A, H, K, L, O, Q, R, U, X	4
C	vor A, H, K, O, Q, U, X außer nach S, Z	4
X	nicht nach C, K, Q	48
L		5
M, N		6
R		7
S, Z		8
C	nach S, Z	8
C	im Anlaut außer vor A, H, K, L, O, Q, R, U, X	8
C	nicht vor A, H, K, O, Q, U, X	8
D, T	vor C, S, Z	8
X	nach C, K, Q	8

Diese Verwendung hat den Nachteil, dass nur eine binäre Entscheidung getroffen wird ob ein Name übereinstimmt oder nicht. Dadurch erhalten wir eine unsortierte Liste mit ähnlichen Namen und da das Verfahren darauf ausgelegt ist möglichst viele Varianten wird die Liste in der Regel sehr lang. Falls aber eine Änderung nicht abgedeckt wird, ist der Name nicht im Ergebnis enthalten. Daher werden wir keinen binären Vergleich der beiden phonetischen Codierungen einsetzen, sondern diese mit einem Ähnlichkeitsmaß für Zeichenketten vergleichen.

Die String-Ähnlichkeit wird mittels des Dice-Koeffizienten (Dice, 1945) der Bigramme der beiden Strings berechnet. Dies ergibt einen Wert im Bereich 0 – 1, wobei eins für eine exakte Übereinstimmung steht.

$$dice(x, y) = \frac{2|bigramme(x) \cap bigramme(y)|}{|bigramme(x)| + |bigramme(y)|} \quad (5.1)$$

5. Verknüpfung mit einer geographischen Datenbank

Für Namen welche die gleiche phonetische Codierung ergeben, erhalten wir hier aber immer noch den gleichen Score. Um hier eine feinere Sortierung nach der Ähnlichkeit zu erreichen, werden wir noch zusätzlich den Dice-Koeffizienten der Originalstrings zum Dice-Koeffizienten der codierten Strings addieren, wie in Gleichung 5.2 angegeben.

$$name_similarity(x, y) = dice(phonetik(x), phonetik(y)) + dice(x, y) \quad (5.2)$$

5.3. Evaluierung

Für die Evaluierung werden die manuell der Ortsdatenbank zugeordneten Namen der Seutter-Karte des Fränkischen Reichskreis A.2.6 verwendet. Die Karte enthält ca. 3800 Ortssignaturen von denen 1626 eine manuelle Zuordnung zu der Ortsdatenbank besitzen. Zum Abgleich werden alle 18720 Orte der Ortsdatenbank welche zwischen dem 48,7 und 50,6 Breitengrad sowie dem 9,1 und 12,5 Längengrad liegen verwendet. Die Performance des Verfahrens wird in zwei verschiedenen Konfigurationen bestimmt. Zum ersten wird immer die vollständige Ortsliste verwendet. Dadurch ist immer ein korrekter aus ca. 18.000 Einträgen zu finden. Je größer die Anzahl der falschen Möglichkeiten ist, desto größer ist auch die Wahrscheinlichkeit, dass einer dieser Einträge eine hohe Ähnlichkeit mit dem Ortsnamen aufweist. In der zweiten Konfiguration werden nur die Orte die in einem gewissen Umkreis um den Zielort liegen, $\pm 0,25^\circ$ Breite und Länge, verwendet.

Der erste Fall stellt eher einen Worst-Case dar, da es viel mehr falsche Kandidaten gibt. Der zweite Fall ist näher an der Verwendung während der Annotation, da hier auch die Position geschätzt wird und nur ein eingeschränkter Teil der Ortsliste verwendet wird. Zur Evaluation der beiden Fälle werden die Metriken *Top-n Accuracy* und *Mean Reciprocal Rank at n* verwendet.

5.3.1. Evaluierungsmetriken und Experimente

Accuracy bezeichnet die durchschnittliche Genauigkeit der Ergebnisse, d. h. die Anzahl der Anfragen bei denen das korrekte Ergebnis zurückgeliefert wurde, oder an der ersten Stelle der Ergebnisliste steht, in Relation zur Gesamtzahl der Anfragen.

Der *Mean Reciprocal Rank* (MRR) ist der Mittelwert des *Reciprocal Rank* (RR) über mehrere Anfragen. Wobei der *Reciprocal Rank* als der Kehrwert der Position des ersten relevanten Treffers in der Ergebnisliste definiert ist. RR ist 1 wenn der erste korrekte Treffer an Position eins steht, wenn der erste korrekte Treffer an Position n der Ergebnisliste auftaucht ist der RR somit $1/n$. (Craswell, 2009)

Mit *Top-n Accuracy* wird die *Accuracy* bezeichnet wenn das korrekte Ergebnis in der Ergebnisliste nicht an der ersten Stelle stehen muss sondern es sich an einer beliebigen der ersten n Stellen befinden kann. $MRR@n$ definiert als der MRR für die auf die ersten n Elemente beschränkte Ergebnisliste.

In den Tabellen 5.2 und 5.3 werden für den alten Ortsnamen „Kolletsham“, welcher den heutigen Ort „Kolitzheim“ bezeichnete, die bis zu 15 besten Ergebnisse pro Verfahren aufgelistet. In Tabelle 5.2 werden alle ca. achtzehntausend modernen Ortsnamen im Bereich zwischen dem 48,7 und 50,6 Breitengrad sowie dem 9,1 und 12,5 Längengrad auf ihre Ähnlichkeit zu „Kolletsham“ hin untersucht und gerankt. In Tabelle 5.3 werden die Ergebnisse für dieselbe Anfrage gezeigt, wobei jedoch die zu vergleichenden Orte auf einen Bereich von $\pm 0,25^\circ$ Breite und Länge um „Kolitzheim“ beschränkt sind, was hier 450 Orten entspricht. Der Ranking-Wert des richtigen Ortsnamens ändert sich hierdurch nicht, dadurch dass hier nur ca. ein vierzigstel der Ortsnamen für das Ranking betrachtet werden müssen, gibt es deutlich weniger falsche Ortsnamen die zufällig eine hohe Übereinstimmung aufweisen und das Ranking wird besser.

In den Abbildungen 5.2a, 5.2b, 5.3a und 5.3b sind die Ergebnisse für die verschiedenen Varianten (eq, dice, phon prefix, phon dice und phon dice + dice) jeweils für die *Top-n Accuracy* und $MRR@n$, im Bereich 1 – 15 für n , dargestellt. Für die Suche in den ca. achtzehntausend Orten, welche die gesamte Kartenfläche abdecken, sind

5. Verknüpfung mit einer geographischen Datenbank

Tab. 5.2.: Resultate für die ähnlichsten Namen zu „Kolletsham“ auf dem gesamten Kartenbereich. Der moderne Ortsname ist „Kolitzheim“.

eq	dice	phon prefix	phon dice	phon dice + dice
	Kollertshof	Geilsheim	Geilsheim	Kolitzheim
	Wolletsthal	Gelsmühle	Gleisenau	Gleußen
	Kollerhammer	Glaszmühle	Gleußen	Külsheim
	Alletshof	Gleisenau	Kaulhausen	Geilsheim
	Ollertshof	Gleisenhof	Klausen	Gleisenau
	Ametshof	Gleismuthhausen	Kolitzheim	Kühholzen
	Allertshausen	Gleitsmühle	Kühholzen	Kaulhausen
	Ellertshausen	Gleizendorf	Külsheim	Eggolsheim
	Hagetshof	Gleißenberg	Egelsheim	Gleitsmühle
	Kollerhof	Gleißenthal	Eggolsheim	Acholshausen
	Klettigshammer	Gleußen	Eglosheim	Kleehaus
	Leupoldshammer	Kaulhausen	Gelsmühle	Kohlhaus
	Alletsried	Klausen	Gickelhausen	Quellitzmühle
	Alletswind	Klausenhof	Glaszmühle	Waigolshausen
	Grametshof	Klausenhäusel	Gleisenhof	Illesheim

Tab. 5.3.: Resultate für die ähnlichsten Namen zu „Kolletsham“ in der Umgebung von „Kolitzheim“. Der moderne Ortsname ist „Kolitzheim“.

eq	dice	phon prefix	phon dice	phon dice + dice
	Ellertshausen	Klesenmühle	Kolitzheim	Kolitzheim
	Weipoltshausen	Kolitzheim	Klesenmühle	Waigolshausen
	Etwashausen		Waigolshausen	Klesenmühle
	Ebertshausen		Alitzheim	Dingolshausen
	Dingolshausen		Dingolshausen	Hergolshausen
	Eckartshausen		Hergolshausen	Weipoltshausen
	Hergolshausen		Mühlhausen	Holzhausen
	Sailershausen		Sulzheim	Fallesmühle
	Volkershausen		Zeilitzheim	Mühlhausen
	Waigolshausen		Fallesmühle	Ellertshausen
	Ballingshausen		Holzhausen	Stolzenmühle
	Happertshausen		Pusselsheim	Pusselsheim
	Hilpertshausen		Reichelshof	Reichelshof
	Hambach		Schloßmühle	Schloßmühle
	Bellevue		Eselsmühle	Alitzheim

die Ergebnisse in Abbildung 5.2 dargestellt und für die auf eine Umgebung von bis zu 0,25° Breite und Länge um den Zielort beschränkte Suche sind die Ergebnisse in Abbildung 5.3 dargestellt. Dies simuliert die Beschränkung auf ein Gebiet durch die für die Gesamtkarte berechnete Georeferenzierung. Folgende Ansätze werden verglichen:

eq Es werden nur moderne Orte, deren Namen exakt mit dem gesuchten Namen übereinstimmt, zurückgegeben.

dice Die modernen Orte werden sortiert nach der Übereinstimmung ihres Namens mit dem gesuchten Namen zurückgegeben, siehe Gleichung 5.1.

phon prefix Die zurückgegebenen Orte werden, wie in Höhn et al. (2013) beschrieben, nach ihrer phonetischen Übereinstimmung gefiltert.

phon dice Die Ortsnamen werden nach der Übereinstimmung der phonetischen Codierung ihrer Ortsnamen sortiert, siehe Gleichung 5.1.

phon dice + dice Die Ortsnamen werden nach der kombinierten Übereinstimmung der phonetischen Codierung ihrer Ortsnamen und der Ortsnamen selbst sortiert, siehe Gleichung 5.2.

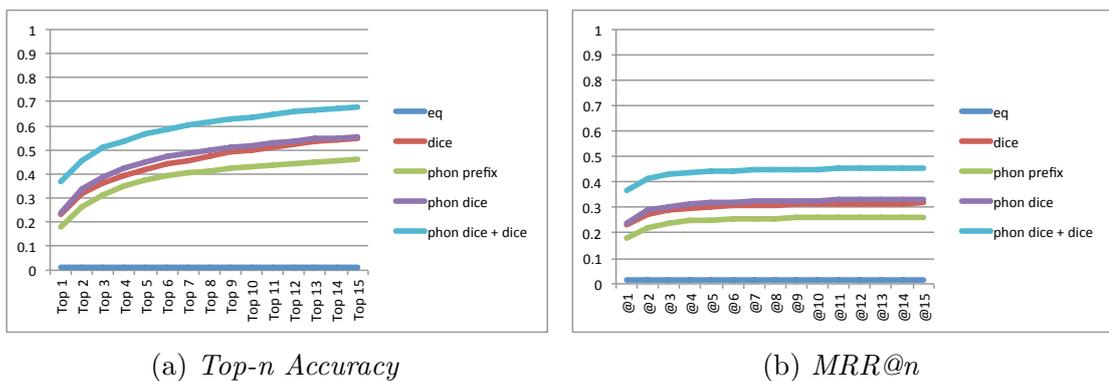


Abb. 5.2.: *Top-n Accuracy* und *MRR@n*, im Bereich 1–15 für n , bei Suche in ca. achtzehntausend Orten, welche die gesamte Kartenfläche abdecken.

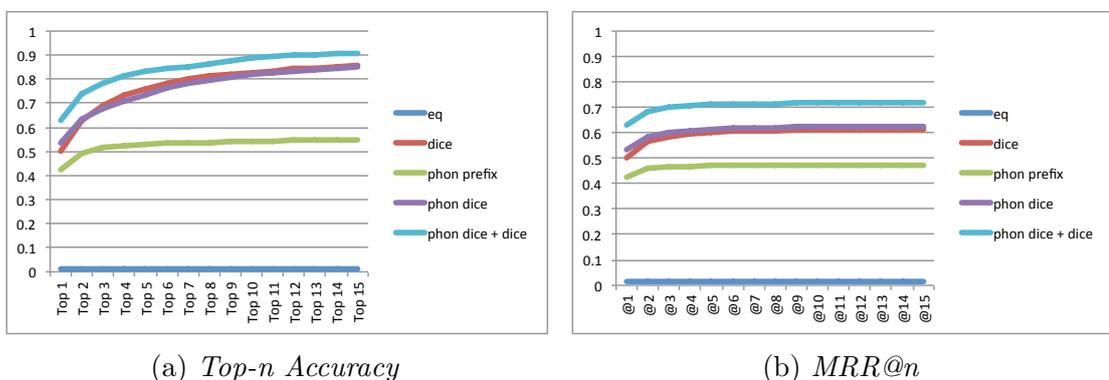


Abb. 5.3.: *Top-n Accuracy* und *MRR@n*, im Bereich 1 – 15 für n , bei Suche in einer Entfernung bis zu 0,25° Breite und Länge um den Zielort. Im Durchschnitt enthält dieses Gebiet 696 Orte.

5. Verknüpfung mit einer geographischen Datenbank

5.3.2. Diskussion der Ergebnisse

Die Ergebnisse in 5.2 und 5.3 zeigen, dass eine sortierte Ergebnisliste besser als ein Filtern der Ergebnisse geeignet ist, um die besten Ergebnisse zu präsentieren. In allen Tests haben die direkte Sortierung nach dem Dice-Score und die Sortierung der phonetischen Codierung nach dem Dice-Score ähnliche und bessere Werte erreicht. Der kombinierte Dice-Score für die phonetischen Codierung und die Namen erreicht die besten Werte, dies lässt sich darauf zurückführen, dass beide Ansätze ihre Stärken in verschiedenen Gebieten haben und diese hier zusammenkommen. Durch den Dice-Score der phonetischen Codierung wird nur eine grobe Sortierung erreicht, es können z. B. viele komplett verschiedene Buchstabenfolgen die gleiche phonetischen Codierung aufweisen, durch die Kombination mit dem Dice-Score des Namens werden diese nun noch nach ihrer Übereinstimmung in der Schreibweise sortiert. Ortsnamen, bei denen die historische und die moderne Schreibweisen übereinstimmen, lagen hier bei ca. 1%, damit eignen sie sich nicht als Suchkriterium.

5.4. Zusammenfassung

In diesem Kapitel wurde eine Lösung vorgestellt, um die dritte Forschungsfrage zu beantworten:

Wie lässt sich die Zuordnung zu modernen Orten erleichtern oder automatisieren (Georeferenzierung)?

Ein neuer Ansatz wurde vorgestellt, um die Zuordnung der Orte einer Altkarte basierend auf seinem Namen und den bereits georeferenzierten Orte der Karte zu erleichtern. Die Experimente haben gezeigt, dass bei der in diesem Kapitel vorgestellte Methode mit 88,9% in den Top-10-Vorschlägen die richtige Zuordnung enthalten ist. Dies reduziert die Georeferenzierung eines Ortes von einer manuellen Suche in der kompletten Geodatenbank auf eine Auswahl aus 10 Orten. Diese können auch noch übersichtlich auf einer Karte dargestellt werden, was die Lage hinsichtlich der weiteren Orte veranschaulicht und somit die Auswahl vereinfacht.

6. Kartenübergreifende Annotationsübertragung

Für moderne Karten oder Luftbilder ist eine Georeferenzierung in einer GIS-Software nach dem Festlegen weniger Kontrollpunkte möglich. Im Gegensatz dazu gibt es für Altkarten eine Vielzahl von Gründen für Verzerrungen, z. B. Ungenauigkeiten während der Vermessung, bei der Zusammenführung von Daten aus verschiedenen Quellen oder auch dem Wunsch, eine visuell ansprechende Kartendarstellung zu erhalten statt einer akkuraten. Trotz der bereits in den Kapiteln 3, 4 und 5 genannten Unterstützungsmöglichkeiten beginnt der Annotationsprozess für jede Karte von Neuem. Da aber aus den Ergebnissen einer einzigen Vermessung mehrere Karten erstellt wurden, und auch Karten von anderen Karten kopiert wurden, sind kartenübergreifende Regelmäßigkeiten zu erwarten, die wir ausnutzen können, um den Annotationsprozess zu vereinfachen, indem Annotationsdaten zwischen den Karten übernommen werden. Klöti (1994, S. 291) führt an, dass Anton Friedrich Büsching (* 1724; † 1793) den Kartenbestand seiner Zeit auf 16.000 Karten schätzte und schon die Enzyklopädie von Krümitz 1793 von 18.000 Karten ausging. Dabei schätzen beide Quellen den Anteil an Karten, die auf Originalaufnahmen zurückgehen auf nur 10% des Kartenbestandes. Damit besteht eine hohe Wahrscheinlichkeit, wenn mehrere Altkarten eines Gebietes untersucht werden, dass einige auf denselben Vermessungsdaten beruhen.

Um von den Ähnlichkeiten Gebrauch zu machen, müssen wir unter den bereits annotierten Karten solche identifizieren, die auf einer ähnlichen Datengrundlage basieren, und eine Korrespondenz zwischen den Ortssignaturen der Karten herstellen. Dadurch können wir eine Karte in Relation zu einer anderen Altkarte und nicht zu einer modernen Karte/Ortsdatenbank georeferenzieren. Wodurch sowohl die Verzerrungen als auch die enthaltenen Orte eine höhere Übereinstimmung

6. Kartenübergreifende Annotationsübertragung

aufweisen und dadurch die Georeferenzierung vereinfachen. In manchen Fällen ist sogar eine automatische Zuordnung der modernen Orte mit Geokoordinaten zu den Ortssignaturen möglich.

Teile dieses Kapitels wurden im Konferenzartikel Höhn und Schommer (2017a) auf der Digital Humanities Conference 2017 publiziert.

6.1. Identifikation korrespondierenden Ortssignaturen

Bevor Annotationen zwischen Karten übertragen werden können, müssen wir geeignete Kartenpaare finden. Voraussetzung hierfür ist, dass in den Karten die Ortssignaturen schon annotiert sind. Hierfür kann eine der in Kapitel 3 beschriebenen Methoden verwendet werden oder die Ortssignaturen können auch manuell erfasst worden sein. Zusätzlich müssen zur Initialisierung der Suche mindestens vier übereinstimmende Punkte zwischen den Karten bekannt sein. Diese können beispielsweise durch den Nutzer vorgegeben werden oder aus den schon mit der Ortsdatenbank verknüpften Orten bestimmt werden.

Der Algorithmus um korrespondierende Ortssignaturen in zwei Karten (A und B) mit M , der Liste von mindestens vier einander zugeordneten Ortssignaturen, zu finden, kann in die folgenden zwei Schritte aufgeteilt werden:

1. *Bestimmen einer Transformation zwischen den Karten.* Die Zuordnung der Koordinaten in M , der Menge von korrespondierenden Ortspositionen zwischen den beiden Karten, werden benutzt um eine projektive Transformation zwischen den Karten A und B zu bestimmen.
2. *Erweiterung der Liste von korrespondierenden Ortssignaturen.* Mithilfe der in Schritt 1 bestimmten projektiven Transformation wird Karte B in das Koordinatensystem von Karte A transformiert. Die so transformierte Karte B bezeichnen wir als B' . Für jede Ortssignatur aus A wird die nächstgelegene Ortssignatur aus B' bestimmt. Wenn es für die Ortssignatur aus B' keine

6.1. Identifikation korrespondierenden Ortssignaturen

näher gelegenen Ortssignatur als die aus A gibt, werden noch folgende Bedingungen geprüft:

- Die Ortssignaturen mit dem zweitkleinsten Abstand müssen mindestens den doppelten Abstand aufweisen wie die zwei betrachteten Ortssignaturen.
- Jede der Ortssignaturen muss durch eine Kante in der Delauny-Triangulierung (Lee und Schachter, 1980) seiner Karte zu einer Ortssignatur aus M verbunden sein.

Wenn diese beiden Bedingungen erfüllt sind, werden diese Ortssignaturen zu M' hinzugefügt. Wenn M' nach der Durchführung von Schritt 2 mehr Elemente als M beinhaltet, dann wird mit Schritt 1 unter Verwendung von M' als M fortgefahren.

M' stellt nun die Zuordnung zwischen den Ortssignaturen der beiden Karten dar. Die einzelnen Schritte des Verfahrens sind in Abbildung 6.2 dargestellt. Die rechte Spalte stellt jeweils die Ergebnisse von Schritt 1 dar und die linke Spalte die von Schritt 2. Die Zeilen entsprechen den verschiedenen Iterationen des Algorithmus. Für die Beispiele wurden die Karten “Nova Franconiae descriptio/Sculptum apud Abrahamum Goos. - Amsterdam: Joannes Janßonius, 1626” (siehe Anhang A.2.4) und “Franckenlandt = Francia orientalis/Per Gerardum Mercatorem – o.O., ca. 1600” (siehe Anhang A.2.3) verwendet.

Die hier verwendeten Karten beinhalten beide ca. 900 Ortssignaturen und überschneiden sich in einem Gebiet welches ca. 800 dieser Ortssignaturen umfasst. Für Goos wurden alle Ortspositionen manuell verifiziert und für Mercator wurden die durch das Template-Matching bestimmten Ortssignaturen verwendet. Hierdurch konnte eine automatische Zuordnung von 755 Ortssignaturen zwischen diese beiden Karten erreicht werden.

Ein anderer Anwendungsfall für diese Abbildung zwischen zwei Karten ist der automatische Vergleich der Ortssignaturen aus zwei Karten. Die Unterschiede zwischen den in zwei Karten annotierten Ortssignaturen können dargestellt werden und erlauben es die Unterschiede in den annotierten Ortssignaturen, egal ob manuell oder automatisch erfasst, auf einfache Weise zu untersuchen. Die identifizierten

6. Kartenübergreifende Annotationsübertragung

Unterschiede heben Bereiche in den Karten hervor die interessant für weitere Untersuchungen sind. Beispiele für solche Unterschiede werden in Abbildung 6.1 gezeigt. Die vier Ausschnitte zeigen großteils die gleichen Daten, weisen aber einige Unterschiede auf. So enthält die Karte in Abbildung 6.1a als einzige den Ort „Hoeltriech“ aber es ist auch die einzige der Karten auf denen „Fuechstat“ und „Aisloben“ nicht abgebildet sind. Durch eine Falte in der Karte wurde auch die Ortssignatur zu „Eßlam“ nicht gefunden und ist als weitere fehlende Ortssignatur markiert. Die Karten 6.1b bis 6.1d stimmen in den dargestellten Orten überein, jedoch mussten in 6.1c durch die breitere Flussdarstellung zwei Orte, am linken Rand, verschoben werden.



(a)



(b)



(c)



(d)

Abb. 6.1.: Darstellung von Unterschieden, rot markiert, in den abgebildeten Orten auf vier Kartenausschnitten.

6.1. Identifikation korrespondierenden Ortssignaturen

Die Gründe für die Unterschiede zwischen den Karten lassen sich in zwei Klassen einteilen. Zum einen kann ein Fehler in der manuellen oder automatischen Annotation vorliegen, was dann hilft diese Probleme zu identifizieren, oder es liegt dem ein wirklicher Unterschied in den Karten zugrunde. Unterschiede zwischen den Karten können selbst natürlich auch wieder auf verschiedenen Gründe zurückgeführt werden, wie z. B. verschiedene Entscheidungen welche Orte wichtiger sind um sie darzustellen oder auch Fehler beim Erstellen der Karten.

In Abbildung 6.2 werden die Ortssignaturen der Goos 1626 (A.2.4) und Mercator 1600 (A.2.3) Karte dargestellt, wenn beide in ein gemeinsames Koordinatensystem, hier das der Goos Karte, transformiert wurden. Dreiecke stehen für Orte aus der Goos Karte und Quadrate für die Mercator Karte. Ausgehend von einigen Übereinstimmungen werden die Ortssignaturen aus der Mercator 1600 Karte in das Koordinatensystem der Goos 1626 Karte transformiert (linke Spalte). Die rechte Spalte zeigt die daraus gewonnen Ortszuordnungen, welche dann im nachfolgenden Schritt, nächste Zeile, für die Bestimmung der neuen Koordinatentransformation verwendet wird. Dreiecke stehen für Ortssignaturen aus Goos und Quadrate für Ortssignaturen aus Mercator. Grüne und rote Marker haben keine zugewiesene Entsprechung in der anderen Karte, im Gegensatz zu den blau und purpurnen.

Die aktuelle Implementierung verwendet eine projektive Transformation um zwei Karten aneinander auszurichten. Dies limitiert die Methode wenn eine lineare Transformation nicht die Verzerrungen auf der kompletten Kartenfläche abbilden kann oder der Inhalt einer Karte aus dem Inhalt mehrerer anderer Karten zusammengesetzt wurde.

Die Korrespondenz von Ortssignaturen welche, in M erfasst ist, macht es möglich einen Teil der Annotationen zwischen den Karten auszutauschen. In unserem Fall in erster Linie die Zuordnung zu einem modernen Ort in der Ortsdatenbank. Andere Informationen wie die Schreibweise können nicht direkt übertragen werden, da diese auf beiden Karten unterschiedlich sein kann. Allerdings kann der Ort über die moderne Schreibweise und alle Varianten die durch Kartenverknüpfungen in der Ortsdatenbank vorhanden sind gefunden werden.

6. Kartenübergreifende Annotationsübertragung

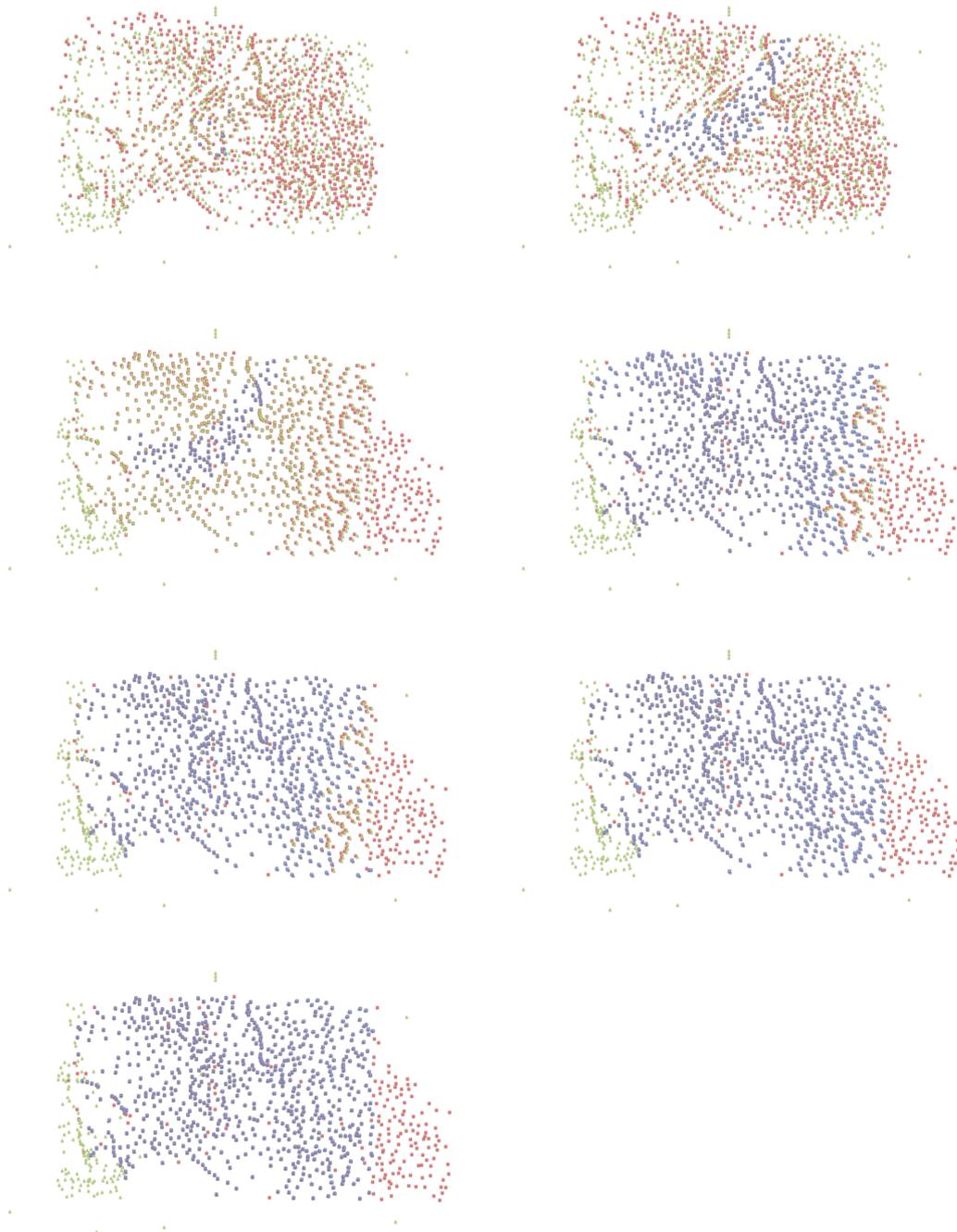


Abb. 6.2.: Darstellung des schrittweisen Vorgehens bei der Ortszuordnung zwischen verschiedenen Karten. Beschreibung siehe Text.

6.2. Evaluierung

In diesem Abschnitt werden die Ergebnisse des in diesem Kapitel vorgestellten Verfahrens zur Identifikation korrespondierender Ortssignaturen ausgewertet.

Es werden mehrere verschiedene Szenarien für den Einsatz des hier vorgestellten Verfahrens betrachtet:

- Übertragen von Annotationen
- Finden von Annotationsfehlern/Unterschieden
- Identifikation welche Karte/Kartenzustand als Grundlage zur Kartenerstellung gedient hat

Diese Szenarien werden empirisch bewertet, um sie miteinander zu vergleichen und deren Vorteile und Nachteile festzustellen. Im Abschnitt 6.2.1 wird der Aufbau der Experimente erklärt und die Szenarien beschrieben. Anschließend im Abschnitt 6.2.2 werden die Ergebnisse diskutiert.

6.2.1. Experimentaufbau

Die Experimente in diesem Teil der Arbeit sind in synthetische Tests und Tests mit realen Kartendaten aufgeteilt. Synthetische Tests werden verwendet, in denen reale Kartendaten mit Fehlern versehen werden, fehlende und zusätzliche Orte sowie Abweichungen der Position eingefügt werden. Es wird somit getestet, ab welcher Fehlerrate eine Zuordnung nicht mehr möglich ist. Dies ist mit realen Kartendaten nicht möglich, da wir diese Fehlerparameter nicht frei wählen können. Daher werden synthetische Tests verwendet, um die Grenzen des Verfahrens abschätzen zu können. Da sich die Eigenschaften der synthetischen Daten nicht vollständig mit realen Daten decken werden, werden auch Tests auf realen Daten durchgeführt, um die Validität des Verfahrens in der Realität zu zeigen und auch um einige der Einsatzszenarien konkret zu zeigen.

6. Kartenübergreifende Annotationsübertragung

Metriken

Zum Bewerten der Zuordnung zwischen zwei Karten werden wir die Metriken *Recall* und *Precision* verwenden.

$$Recall = \frac{|\{\text{Übereinstimmende Orte}\} \cap \{\text{Einander zugeordnete Orte}\}|}{|\{\text{Übereinstimmende Orte}\}|} \quad (6.1)$$

$$Precision = \frac{|\{\text{Übereinstimmende Orte}\} \cap \{\text{Einander zugeordnete Orte}\}|}{|\{\text{Einander zugeordnete Orte}\}|} \quad (6.2)$$

Der *Recall* gibt an wie viele der Übereinstimmungen aus beiden Karten im Ergebnis enthalten sind und *Precision* gibt an wie hoch der Anteil an richtigen Ergebnissen in der Ergebnismenge ist.

Tests

Für die Tests werden zwei Scans von verschiedenen Abdrucken, einer Kupferplatte benutzt, siehe Anhang A.1.5. Die Karten enthalten 598 Ortssignaturen, die in beiden Karten vollständig erfasst sind. Wenn die zweite Karte mittels einer projektiven Transformation in das Koordinatensystem der ersten Karte abgebildet wird, so besteht ein durchschnittlicher Abstand von 5,8 Pixeln von den projizierten zu den korrespondierenden Punkten. Bei einem Durchmesser von ca. 18 Pixeln für den Kreis in der Ortssignatur und einen minimale Abstand von ca. 35 Pixeln zwischen zwei Ortssignaturen. Ein Histogramm für die jeweils minimalen Abstände zur nächstgelegenen Ortssignatur wird in Abbildung 6.3 gezeigt.

Um die Robustheit der Zuordnung zu testen, werden in einer der Karten Fehler hinzugefügt. Zum einen werden die Ortspositionen zufällig verschoben, hierzu wird eine standardnormalverteilte Zufallszahl d mit einem Faktor f multipliziert um die Distanz zu bestimmen und eine gleichverteilte Zufallszahl α aus dem Intervall $[0, \pi[$ bestimmt die Richtung der Verschiebung.

$$\vec{v}_{disp} = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix} df \quad (6.3)$$

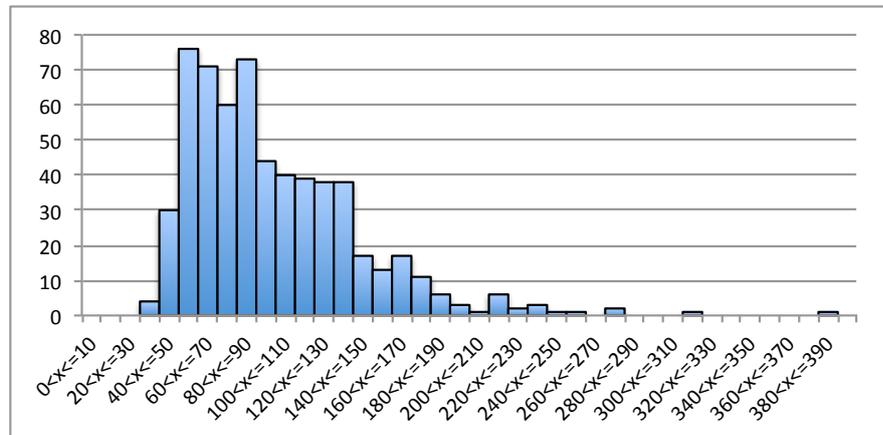


Abb. 6.3.: Histogramm der jeweils minimalen Abstände zur nächstgelegenen Ortssignatur.

Jede Ortsposition wird zuerst um einen jeweils Zufällig bestimmten Vektor \vec{v}_{disp} verschoben. Dies simuliert Fehler beim exakten platzieren der Ortssignaturen auf der Karte. Zusätzlich werden Fehler bei der Erfassung der Ortssignaturen simuliert. Hierzu werden einige zufällig ausgewählte Orte entfernt, wobei als übereinstimmend zwischen den Karten markierte Orte nicht entfernt werden, und auch neue hinzugefügt. Hierdurch könne verschiedene *Recall*- und *Precision*-Werte simuliert werden. In den Karten sind acht übereinstimmende Ortspositionen markiert, welche als Startpunkte für die weitere Zuordnung von Ortspunkten genutzt werden. Diese sind in Abbildung 6.4 hervorgehoben, wobei die Ortspositionen in der zweiten Karte zufällig, nach Gleichung 6.3 mit einem Faktor $f = 15$, verschoben wurden.

Um die Auswirkungen von ungenauen Platzierungen von Ortssignaturen auf den Algorithmus zur Ortszuordnung zu prüfen, wurden verschiedene Verschiebungsfaktoren im Bereich 0 – 100 getestet. Die Ergebnisse von vier Simulationsläufen hierzu sind in Abbildung 6.5 zusammengefasst. Aus der Grafik lässt sich erkennen, dass bei kleinen Verschiebungen der *Recall* bei 100% bleibt, dann kontinuierlich abfällt bis ein Wert von ca. 50% erreicht ist um dann fast direkt auf Werte von unter 10% abzufallen. Der Wert für *Precision* ist bei allen Versuchen konstant auf 1.0 verblieben, was bedeutet, dass es zu keinen falschen Zuordnungen kam. Wenn der *Recall* ca. 50% erreicht sind die Orte durchschnittlich schon um mehr als 25 Pixel verschoben.

6. Kartenübergreifende Annotationsübertragung

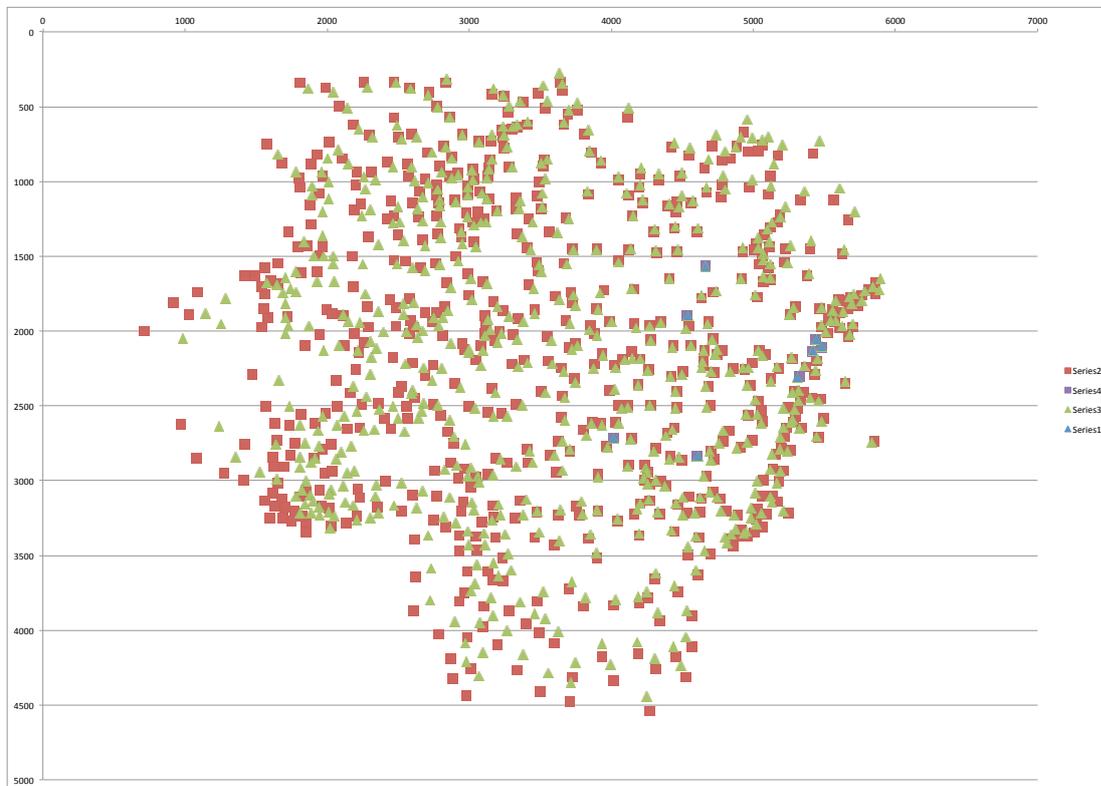


Abb. 6.4.: Verzernte Karte 2 in das Koordinatensystem von Karte 1 projiziert. Die Projektion wurde mithilfe der acht zugeordneten Orte berechnet.

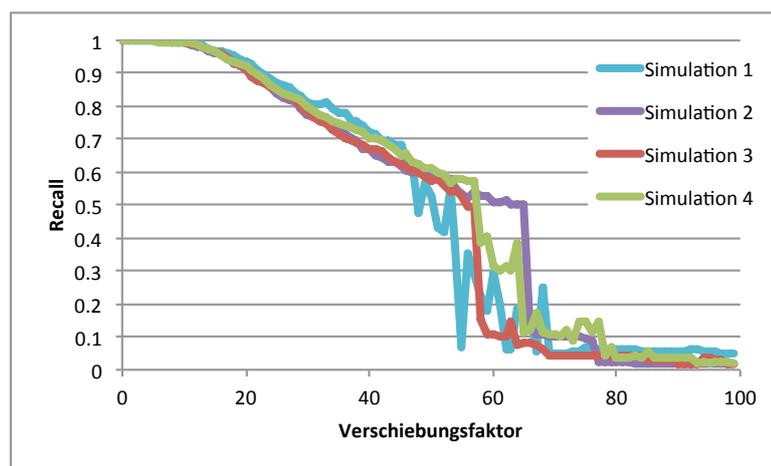


Abb. 6.5.: *Recall* in Abhängigkeit von verschieden starken zufälligen Verschiebungen in den Ortsplatzierungen.

Um auch die Auswirkungen von Fehlern während der Erfassung oder durch Unterschiede bei der Erstellung einer Karte zu bestimmen, wurden für diese Karte auch die Auswirkungen von fehlenden und hinzukommenden Orten untersucht. Fehlende Orte können ihren Ursprung entweder im Fehlen des Ortes in der zu vergleichenden Karte oder einem Erfassungsfehler haben, entsprechend für hinzugekommene Orte. In Abbildung 6.6 sind die Ergebnisse hierzu zusammengefasst. Die Simulationen wurden für *Recall* und *Precision* in den Ausgangsdaten mit den Werten von $[0,05 - 1]$ in Schritten von 0,05 durchgeführt. In den Zeilen sind jeweils Ergebnisse für verschiedene Verschiebungsfaktoren f , aufsteigend von 0 bis 50 in Inkrementen von 10, dargestellt. Links ist jeweils der *Recall* und rechts die *Precision* der automatischen Zuordnung von Orten in Abhängigkeit von *Recall*, *Precision* in den Ausgangsdaten dargestellt. In Abbildung 6.6a lässt sich der *Recall* für die Ortszuordnung sehen, wenn die Ortspositionen nicht zusätzlich verschoben wurden. Für einen *Recall* $> 0,5$ in den Ausgangsdaten wird bei der Zuordnung mindestens ein *Recall* von 0,72 erreicht und bei einem *Recall* $> 0,6$ in den Ausgangsdaten mindestens ein *Recall* von 0,93. Für *Precision* $> 0,5$ in den Ausgangsdaten liegt die *Precision* auch immer über 0,92. Wie sich in der rechten Spalte von Abbildung 6.6 erkennen lässt, bleibt die *Precision* für einen großen Bereich bei weit über 0,9.

Zusätzlich wurde der Algorithmus auf einer Serie von Karten getestet welche von einer Kupferplatte, die aber im Laufe der Jahre mehrmals aktualisiert wurde, gedruckt wurden. Diese Karten sind im Anhang A.1.5 beschrieben. In Fritzen und Solchenbach (2017) werden sieben Zustände dieser Karte beschrieben (Tabelle 6.1) und zu sechs dieser Zustände ($Z1 - Z6$) standen Scans für eine Analyse zur Verfügung.

In Abbildung 6.7 ist die Überlagerung der Orte aus den Zuständen $Z1$ und $Z2$ im Koordinatensystem von $Z1$ dargestellt. Es konnte für jeden Ort aus Zustand $Z1$ der entsprechende Ort aus Zustand $Z2$ zugeordnet werden, zusätzlich sind 48 neue Ortssignaturen in Zustand $Z2$ hinzugekommen. In Fritzen und Solchenbach (2017) werden 50 Änderungen, siehe Tabelle 6.1, zwischen diesen Zuständen angegeben, wobei hier nicht nur hinzugekommene Ortssignaturen sondern auch hinzugefügte Waldstücke, Flüsse und Bezeichnungen gezählt wurden. Wenn für jede der angegebenen Kategorien mindestens eine Änderung vorkommt, müsste sich die tatsächliche Zahl der Änderungen auf über 50 belaufen.

6. Kartenübergreifende Annotationsübertragung

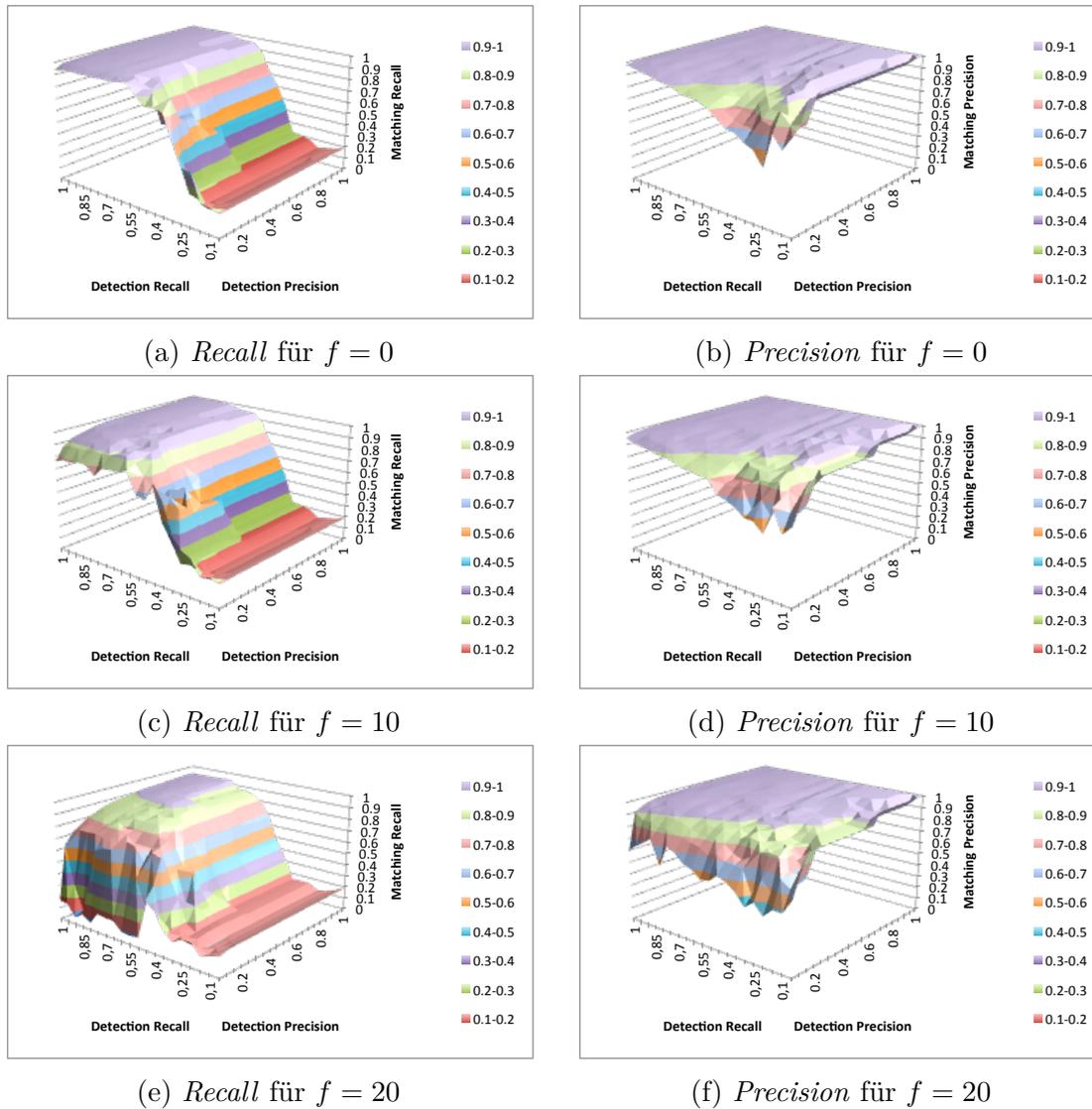


Abb. 6.6.: *Recall* (links) und *Precision* (rechts) der automatischen Zuordnung von Orten in Abhängigkeit von *Recall*, *Precision* und zufälligen Verschiebungen (Zeilen) in den Ausgangsdaten.

Für den Zustand $Z6$ sind in Fritzen und Solchenbach (2017) keine Änderungen in den Ortssignaturen im Vergleich zu Zustand $Z5$ angegeben, es wird nur das Hinzufügen des Textes *Theodorus Galle excudit Antwerpiae* unterhalb der Kartusche aufgelistet. Dies ist auch die einzige Änderung die van der Vekene und van den Broecke in ihren jeweils letzten beschriebenen Zuständen angeben. Die Zuordnung der Ortssignaturen von $Z5$ und $Z6$ zeigt aber zwei neu hinzugekommene Ortssigna-

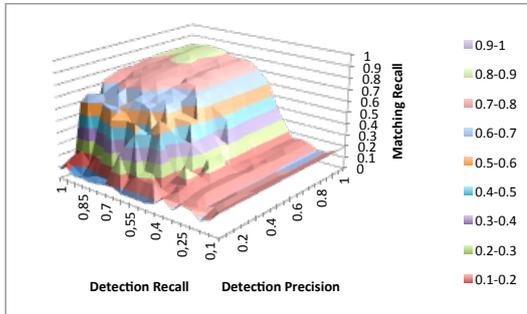
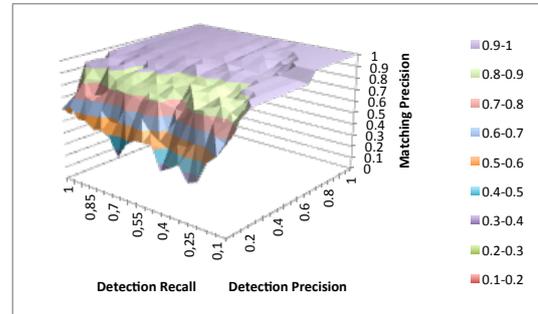
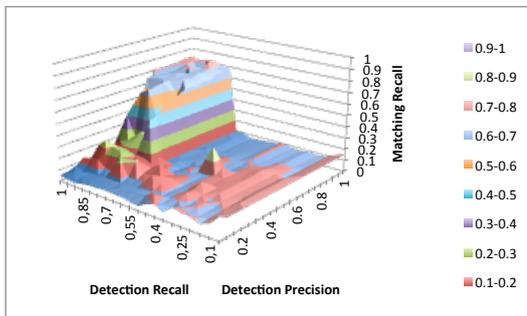
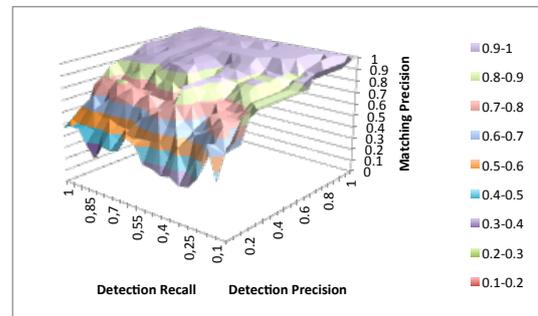
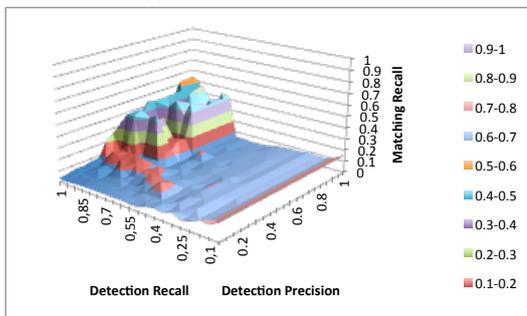
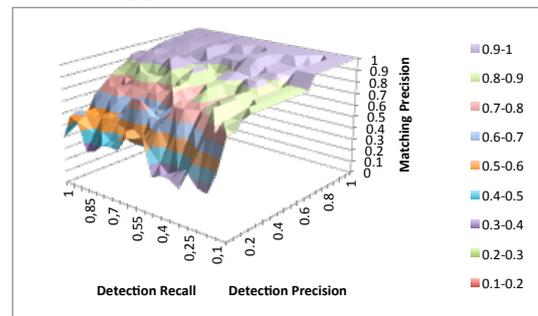
(g) *Recall* für $f = 30$ (h) *Precision* für $f = 30$ (i) *Recall* für $f = 40$ (j) *Precision* für $f = 40$ (k) *Recall* für $f = 50$ (l) *Precision* für $f = 50$

Abb. 6.6.: *Recall* (links) und *Precision* (rechts) der automatischen Zuordnung von Orten in Abhängigkeit von *Recall*, *Precision* und zufälligen Verschiebungen (Zeilen) in den Ausgangsdaten. (fortges.)

turen, zu sehen in Abbildung 6.8 (grün), in Zustand $Z6$. Eine Gegenüberstellung der entsprechenden Kartenausschnitte aus Zustand $Z5$ und $Z6$ ist in Abbildung 6.9 zu sehen. Es besteht hier die Möglichkeit, dass diese Änderungen sowohl in van der Vekene, van den Broecke und Fritzen und Solchenbach (2017) übersehen wurden, oder dass ein weiterer Zustand existiert, der nur den hinzugefügten Text, nicht aber die zusätzlichen Orte, enthält.

6. Kartenübergreifende Annotationsübertragung

Tab. 6.1.: Sieben Zustände der Luxemburg-Karte und ihre Zuordnung zu den publizierten Zuständen bei van der Vekene und van den Broecke. (Fritzen und Solchenbach, 2017)

Zustand	Nr. Vd Boecke	Nr. Vd Vekene	Datierung	Beschreibung der Änderungen	Referenzexemplare in der BNL
Z1	60.1	1.02.A	1579		2695 / IX-A-1
Z2		1.02.B	1581	Es sind 50 neue Einträge (Orte, Waldstücke, Flüsse, Bezeichnungen, etc.) hinzugefügt.	2706 / IX-A-1
Z3	60.2		1584	Es sind 37 neue Einträge dazugekommen.	sans cote
Z4	60.3		1587	Es sind weitere 7 neue Einträge dazugekommen.	sans cote
Z5	60.4	1.02.C	1592	Die Kartusche links unten und einige Signaturen der Wälder und Flüsse sind neu gestochen.	1074 / IX-A-1
Z6	60.5	1.02.d	1612	<i>Theodorus Galle excudit Antwerpiae</i> wurde unterhalb der Kartusche hinzugefügt.	sans cote
Z7				Die Kartusche links unten und einige Signaturen der Wälder und Flüsse sind wiederum neu gestochen.	

Die weiteren Vergleiche der aufeinanderfolgenden Zustände ergaben:

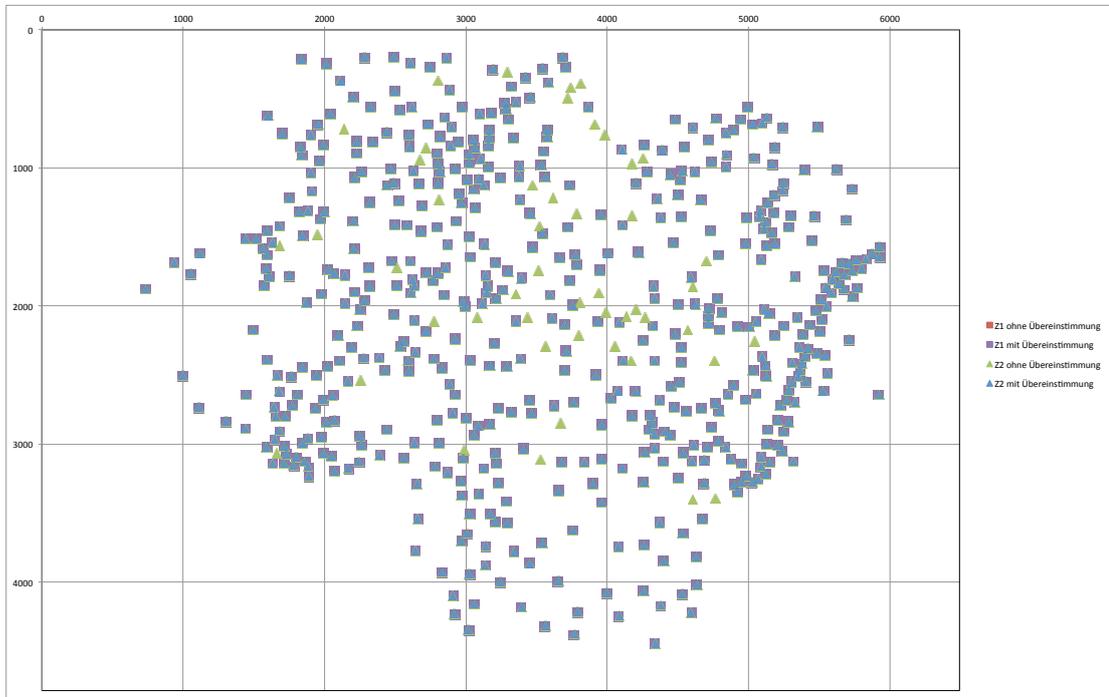


Abb. 6.7.: Darstellung der Unterschieden die mit der vorgestellten Methode in den Zuständen Z1 und Z2 der Karte A.1.5 gefunden wurden. Es wurden 48 zusätzliche Ortssignaturen gefunden (grün) und keine Ortssignatur wurde entfernt (rot).

Zustände	Hinzugekommene Ortssignaturen	Abbildung
Z2 und Z3	18	A.5
Z3 und Z4	7	A.6
Z4 und Z5	0	A.7

Auch hier sind die sieben hinzugekommenen Ortssignaturen zwischen Zustand Z3 und Z4 interessant. Van den Broecke listet auf der Webseite www.orteliusmaps.com (van den Broecke, 2018) alle Orte der Luxemburg-Karte von Ortelius mit dem Zeitpunkt des Hinzufügens auf. Hier sind fünf Orte aufgelistet die zwischen Zustand 60.2 und 60.3 hinzugefügt wurden. Die zwei verbleibenden Orte sind bereits für Zustand 60.2 aufgelistet. Dies kann ein Fehler in der Auflistung sein oder die Zustände Z3 und 60.2 entsprechen einander nicht, sondern stellen verschiedene Zustände der Karte dar.

6. Kartenübergreifende Annotationsübertragung

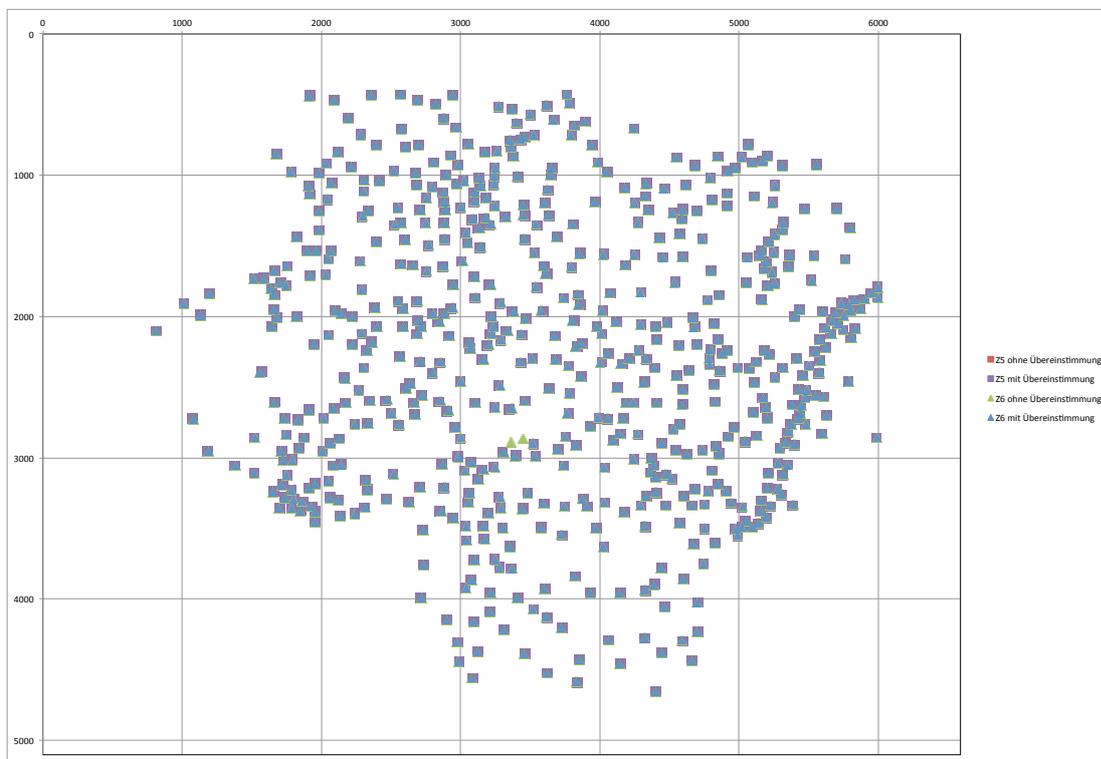


Abb. 6.8.: Darstellung der zwei Unterschiede die mit der vorgestellten Methode in den Zuständen Z5 und Z6 der Karte A.1.5 gefunden wurden. Es wurden zwei zusätzliche Ortssignaturen gefunden (grün) und keine Ortssignatur wurde entfernt (rot).

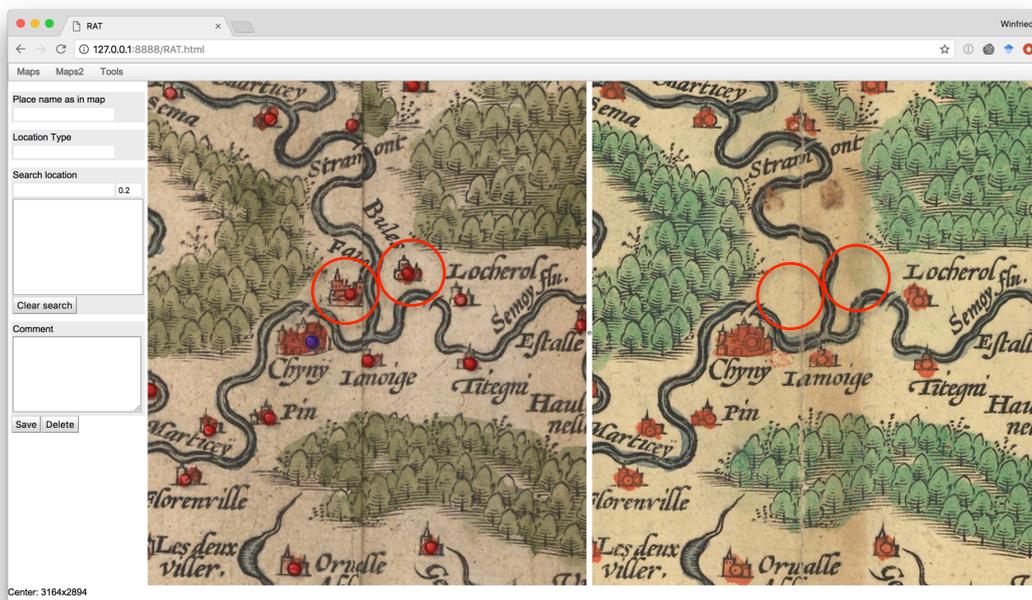


Abb. 6.9.: Darstellung der zwei Unterschiede (rote Kreise) die mit der vorgestellten Methode in den Zuständen Z5 (rechts) und Z6 (links) der Karte A.1.5 gefunden wurden.

6. Kartenübergreifende Annotationsübertragung

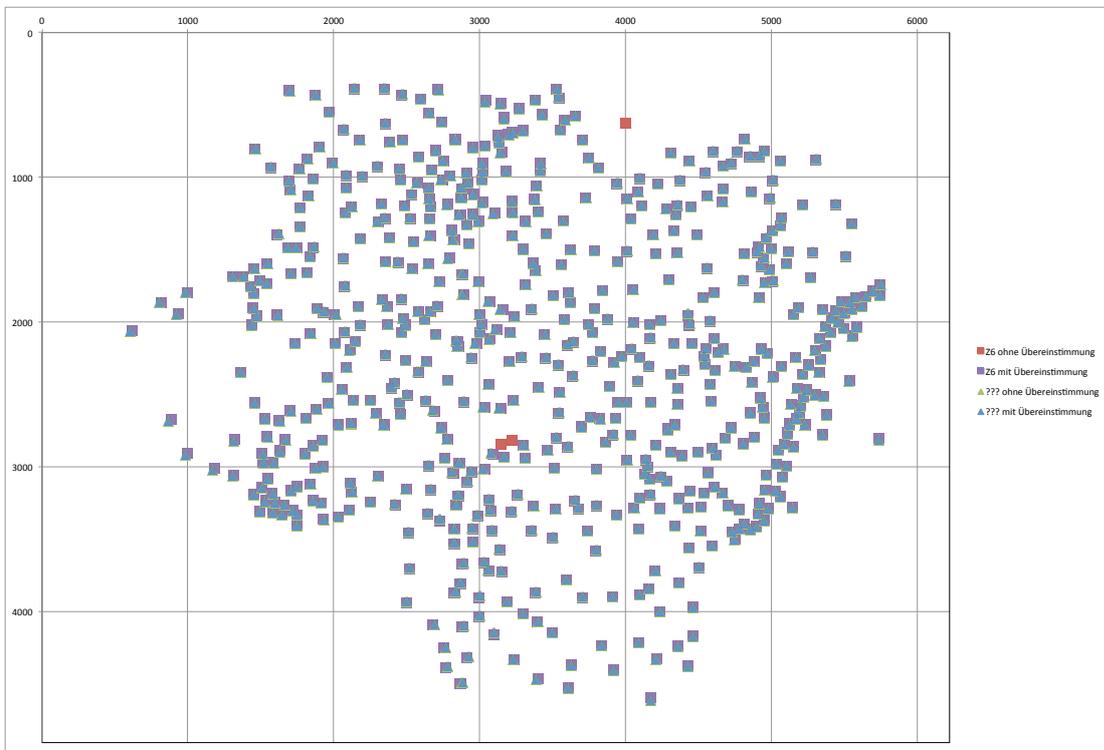


Abb. 6.10.: Darstellung der drei Unterschiede die mit der vorgestellten Methode zwischen dem Zustand Z6 der Ortelius 1579 Karte A.1.5 und der Waesberge 1636 Karte A.1.2 gefunden wurden.

6.2.2. Diskussion

Die in diesem Kapitel beschriebenen Resultate können genutzt werden, um in ähnlichen Karten die Erkennung von Ortssignaturen zu verbessern und Unterschiede zwischen den Karten hervorzuheben. Alle Unterschiede, die sich in überlappenden Kartenbereichen ergeben, sind entweder Unterschiede der Karten oder Fehler in der Erkennung von Ortssignaturen. Bei einem Test mit zwei Karten, welche einen großen Bereich mit übereinstimmenden Ortssignaturen enthalten und leicht unterschiedliche Ost-West-Ausdehnungen haben und daher auch zusätzliche Bereiche ohne Übereinstimmungen besitzen, wurden für ca. 5% der im übereinstimmenden Gebiet erkannten Ortssignaturen keine Zuordnung gefunden. Die manuelle Kontrolle dieser Unterschiede ergab, dass davon ca. 0,5% auf Unterschieden in den Karten basieren und der Rest auf Fehlern in der Erkennung der Ortssignaturen. Durch diese manuelle Kontrolle erhalten wir zwei Karten mit nahezu fehlerfrei erkannten Ortssignaturen und zusätzlich auch eine Auflistung aller Unterschiede der Karten, was sonst der sprichwörtlichen Suche nach der Nadel im Heuhaufen gleichen würde.

Die verknüpften Ortssignaturen könnten sich auch für andere Verarbeitungsschritte als hilfreich erweisen, wie z. B. für die Texterkennung. Durch die Verknüpfung wären schon die in den anderen Karten verwendeten Ortsnamen bekannt, und dies könnte zur Entdeckung von OCR-Fehlern genutzt werden.

Dieser Teil der Arbeit zeigt, dass es möglich ist, eine Zuordnung zwischen den Ortssignaturen zweier Karten herzustellen. Der Aufwand würde dabei dem Aufwand des einfachen Georeferenzierens einer Karten entsprechen. Allerdings würde ein einfaches Georeferenzieren nur die von der Karte abgedeckte Region als Information liefern. Voraussetzung für die kartenübergreifende Annotation ist natürlich eine bereits in der Datenbank vorhandene ähnliche Karte, welche die gleiche Region abdeckt. In der Datenbank kann dann diese ähnliche Karte identifiziert und die sich entsprechenden Ortssignaturen bestimmt werden. Dadurch können dann die Georeferenzierungen einzelner Ortssignaturen von einer Karte auf eine andere übertragen werden. Die Unterschiede in den Mengen der Ortssignaturen können erfasst werden.

6.3. Zusammenfassung

In diesem Kapitel wurde eine Lösung vorgestellt, um die vierte Forschungsfrage zu beantworten:

Wie lässt sich die Ähnlichkeit zwischen Karten nutzen um die Erfassung der Orte zu unterstützen?

Es wurde ein neuer Ansatz vorgestellt, um ähnliche Karten automatisch zu finden. Die einander entsprechenden Ortssignaturen in diesen Karten werden einander zugeordnet. Hierdurch können Annotationen zwischen diesen Karten übertragen werden und der Annotationsaufwand erheblich reduziert werden.

Durch die hergestellte Verknüpfung der Ortssignaturen ergeben sich noch weitere Möglichkeiten. Dies kann zur Erkennung von Fehlern in der automatischen Ortssignaturerkennung oder der manuellen Annotation genutzt werden. Des Weiteren könne auf diese Weise auch Unterschiede zwischen Kopien von Karten ermittelt werden.

7. RAT - Referenzierungs- und Annotations-Tool

Die in den Kapiteln 3 bis 6 beschriebenen Lösungen sollen im Referenzierungs- und Annotationstool (RAT) vereint werden, um eine einheitliche Oberfläche zu bieten um die Kartenannotationen mit möglichst kleinem manuellem Aufwand zu erfassen. Besonders das Ausnutzen der Informationen aus ähnlichen Karten könnte sich hier in vielen Bereichen als hilfreich zeigen. Frühere Versionen des Systems wurden in Höhn et al. (2013); Höhn und Schommer (2016a,b, 2017b) beschrieben.

RAT bietet eine automatische Erkennung von Ortssignaturen, Ortslabels und unterstützt zusätzlich die Georeferenzierung. Jedem Ort auf der Karte können vier Informationstypen zugeordnet werden: die auf der Karte vorhandene Ortsbezeichnung, die Art des Ortes (z. B. Dorf, Stadt oder Kloster), eine Referenz zu dem Ort in einer modernen Karte und ein Kommentar. Diese Funktionen werden in den folgenden Abschnitten erläutert.

RAT ist als Web-App realisiert. Diese Implementierung erlaubt eine gemeinschaftliche Annotierung, und Teile des Systems können wiederverwendet werden, um die annotierten Karten online zu präsentieren.

7.1. Ortsdaten

Die für die Experimente verwendeten Geodaten stammen von der National Geospatial-Intelligence Agency (NGA), einer US-amerikanischen Behörde für Geographische

7. RAT - Referenzierungs- und Annotations-Tool

Aufklärung. Die NGA sammelt weltweit, mit Ausnahme der USA, raumbezogene Daten und wertet diese aus. Ein Teil dieser Daten wird auch zur freien Verwendung zur Verfügung gestellt, so z.B. die Digital Terrain Elevation Data, ein digitales Geländemodell, welches Höhendaten in einem 1km Raster enthält¹, die Vector Map (VMAP) Daten², vektorbasierte Kartendaten und den GEONet Names Server (GNS), eine Datenbank von Namen und Koordinate geographischer Merkmale³.(NGA About, 2018)

Die GNS Daten stehen als *Country Files*, aufgeteilt nach Ländern oder als weltweites Gesamtpaket, zum Download⁴ bereit. Die jeweiligen Länderdaten enthalten den Datensatz in einer Tab-Getrennten-Datei. Eine Beschreibung der einzelnen Felder ist Online verfügbar⁵.

Für die in RAT verwendete Ortsdatenbank werden nur die Geokoordinaten und Ortsnamen verwendet, hierzu werden nur die Felder *LAT*, *LONG*, *FC* und *FULL_NAME_RO* benötigt, alle anderen der über 30 Felder werden nicht beachtet. Das *FC*-Feld (Feature Class) wird verwendet um auszuwählen welche Daten in die Ortsdatenbank übernommen werden, nur für Zeilen welche hier ein *P* enthalten werden die Geokoordinaten und der Name aus den *LAT*, *LONG* und *FULL_NAME_RO* Feldern übernommen. Der Wert *P* im *FC*-Feld steht für bewohnte Orte.

Es wurden die Ortsdaten der *Country Files* für Deutschland, Luxemburg, Frankreich und Belgien verwendet um die RAT Ortsdatenbank für die in dieser Arbeit durchgeführten Experimente zu füllen.

¹<https://www.nga.mil/ProductsServices/TopographicalTerrestrial/Pages/DigitalTerrainElevationData.aspx>

²<http://earth-info.nga.mil/publications/vmap0.html>

³<http://geonames.nga.mil/gns/html/>

⁴<http://geonames.nga.mil/gns/html/namefiles.html>

⁵http://geonames.nga.mil/gns/html/gis_countryfiles.html

7.2. Benutzeroberfläche

Zuerst erfolgt der Upload eines Kartenscans, dieser wird dann automatisch für die Verwendung in RAT vorbereitet. Dazu wird das Bild in Kacheln von 256x256 Pixeln zerlegt, welche dann für die Onlinedarstellung verwendet werden.

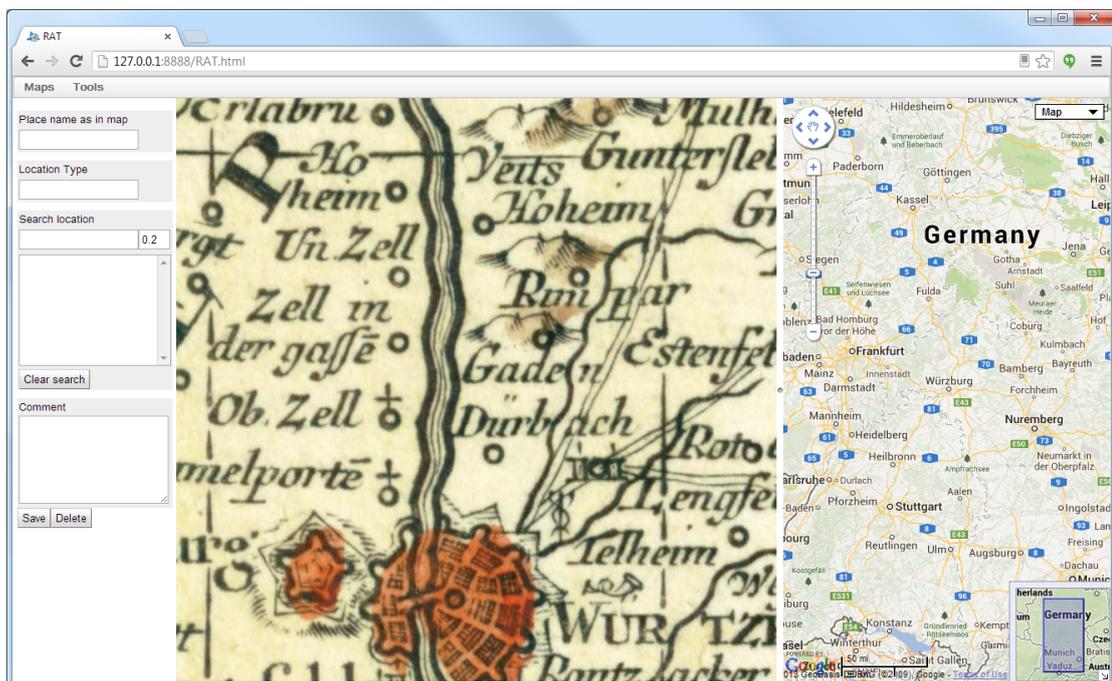


Abb. 7.1.: RAT mit einer neu importierten Karte.

In Abbildung 7.1 ist das Benutzerinterface von RAT nach dem Import einer neuen Karte zu sehen. Es sind Felder vorhanden um die Schreibweise eines Ortsnamens auf der Karte zu erfassen, den Ortstyp, wie z. B. Dorf, Stadt oder Kloster, sowie die Geokoordinaten des Ortes durch eine Verknüpfung zu dem modernen Ort in einer Datenbank und ein Kommentarfeld.

Der nächste Schritt besteht dann aus der Erfassung der verwendeten Ortssignaturen. Diese werden als Muster für das Template-Matching benötigt und können entweder direkt in der Karte erfasst werden oder aus der Kartenlegende, falls diese vorhanden ist. Diese Templates werden wie in Abschnitt 3.2 beschrieben zum Template-Matching verwendet und Maxima über einem Schwellwert werden als Ortssignaturen vorgeschlagen und mit dem beim Muster spezifizierten Typ

7. RAT - Referenzierungs- und Annotations-Tool

versehen.

Um die automatische Erkennung von Ortssignaturen auszuführen muss der Benutzer ein Template für die zu suchenden Ortssignatur angeben. Hierzu muss ein rechteckiger Bereich mit der Ortssignatur ausgewählt werden und der Name und Typ der Ortssignatur angegeben werden. Das System verwendet dann einen Template-Matching-Algorithmus um in der Karte Stellen mit ähnlichen Symbolen zu finden.

Durch die Heterogenität der Ortssignaturen in Altkarten ist es nicht einfach eine hohe Genauigkeit bei der automatischen Erkennung zu erreichen und daher ist eine händische Nachkorrektur nötig. Daher können die automatisch gefundenen Ortssignaturen und deren Typ bearbeitet werden. Nicht erkannte Ortssignaturen können auch von Hand hinzugefügt werden.

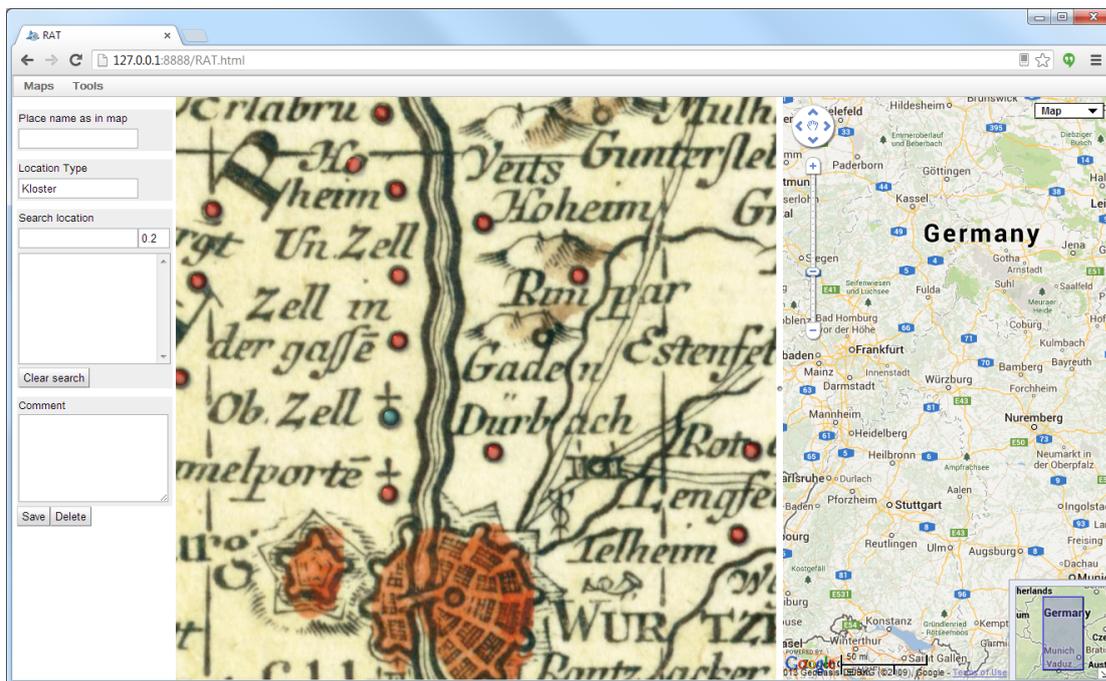


Abb. 7.2.: Gefundene Ortssignaturen.

Als Ergebnis erhält man eine Karte mit den automatisch gefundenen Ortssignaturen. In Abbildung 7.2 ist dies zu sehen und für die ausgewählten Ortssignatur (blau) wird auch der automatisch eingetragene Typ gezeigt. Auch sind links die Felder für die zusätzlichen Annotationen, historischer Name, Georeferenz und Kommentar

zu sehen.

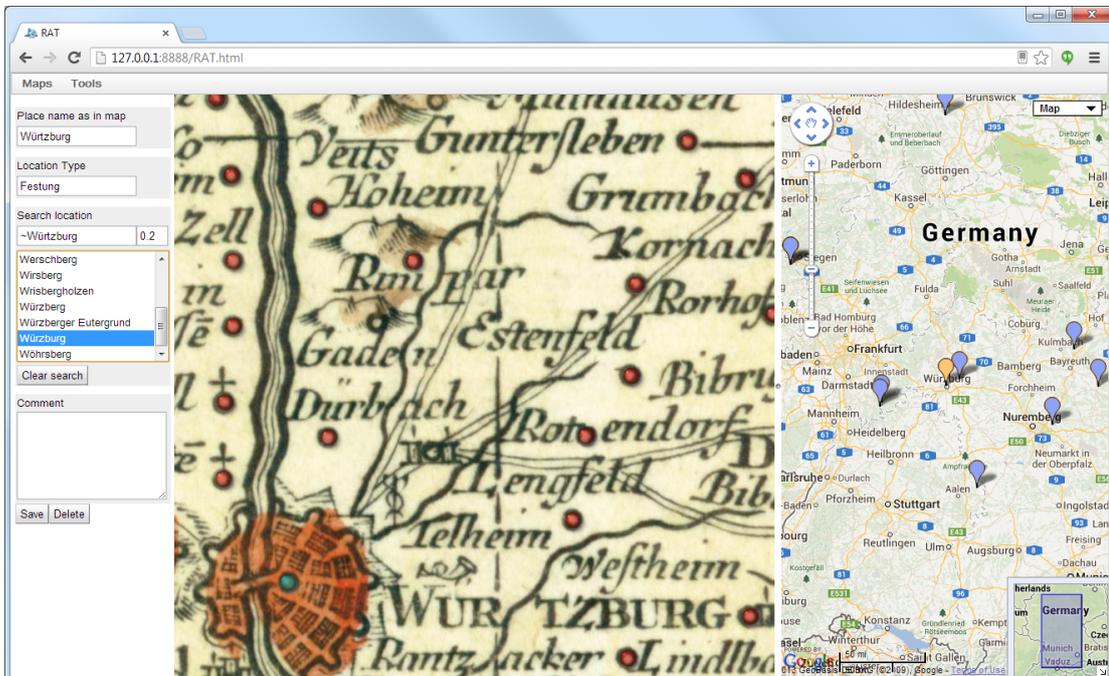


Abb. 7.3.: Hinzufügen der ersten Ortssignatur.

Als Ergebnis erhält man eine Karte mit den automatisch gefundenen Ortssignaturen. Dies ist in Abbildung 7.3 zu sehen. Außerdem wird das Hinzufügen des ersten Ortssignatur gezeigt. Bei diesem sind noch nicht ausreichend Informationen für eine Abschätzung der Geoposition vorhanden, hierfür werden sechs eingemessene Punkte benötigt. Die phonetische Suche steht für diesen ersten Punkt aber natürlich schon zur Verfügung. Die phonetische Suche ist hilfreich da sich die Schreibweise von Ortsnamen mit der Zeit geändert hat.

Nach dem sechsten eingemessenen Ortspunkt steht dann auch die Positionsvorhersage zur Verfügung und kann den Suchbereich für einen passenden Ort eingrenzen 7.4. Auch ist es nicht nötig den Ortsnamen in seiner Schreibweise auf der Karte und für die Suche zweimal einzugeben, die Kartenschreibweise wird automatisch für die Suche übernommen. In diesem Beispiel wird „Telheim“ von der Kartenschreibweise übernommen und es werden in der Nähe der vorhergesagten Ortsposition zwei Orte, welche beide „Theilheim“ heißen, gefunden.

In einigen Fällen ist die Zuordnung ohne weiteres Wissen nicht möglich, wie im

7. RAT - Referenzierungs- und Annotations-Tool

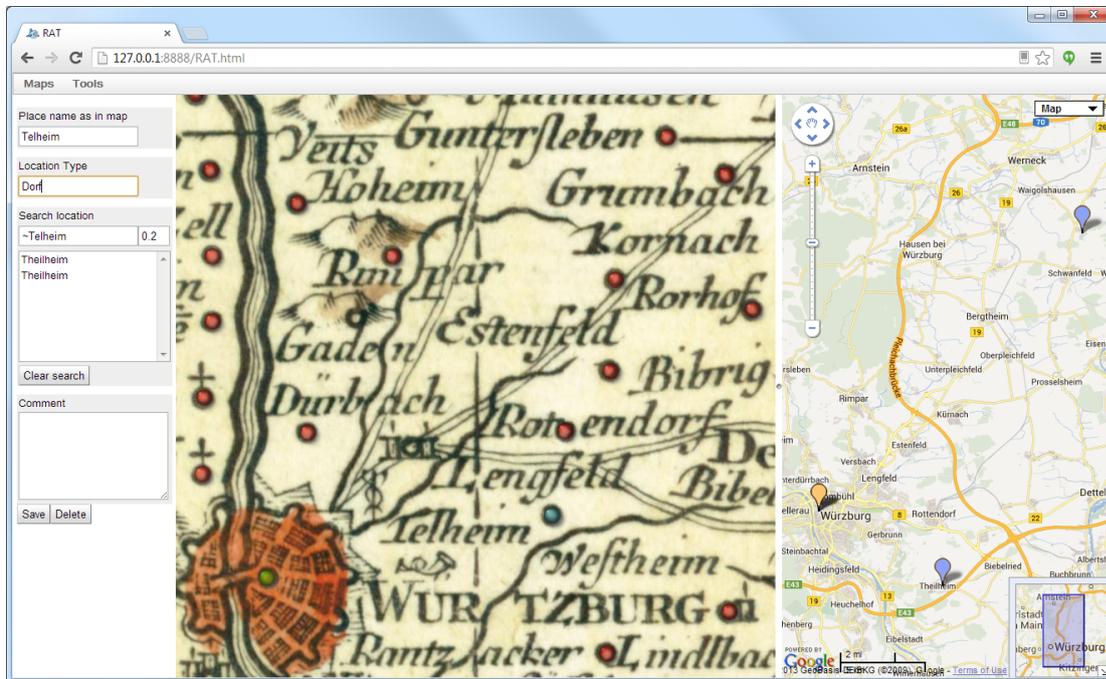


Abb. 7.4.: Hinzufügen weiterer Ortssignaturen.

Beispiel auf Abbildung 7.5. Zu dem markierten Ort „Dürbach“ werden die zwei Orte „Oberdürbach“ und „Unterdürbach“ gefunden, welche auch anhand ihrer Position nicht zugeordnet werden können.

In Abbildung 7.6 wird die Annotation eines Textlabels gezeigt. Hierzu wird zuerst die Grundlinie markiert indem, nach Aktivierung des Textannotations-Modus zuerst der Beginn angeklickt und der Mauszeiger bis zum Ende der Grundlinie gezogen wird. Anschließend wird der Mauszeiger bis auf Höhe der x-Linie bewegt und die Annotation durch einen Klick bestätigt. Dies kann mit der Texterkennung verknüpft werden, hierdurch wird für jedes annotierte Textlabel kann eine automatische Texterkennung durchgeführt. Das Ergebnis wird als Transkription des Labels vorgeschlagen, als Ortsname eingetragen und die Ortssuche wird ausgelöst. Hierdurch muss im besten Fall nur noch der passende moderne Ort aus den Vorschlägen gewählt werden.

Diese automatische Texterkennung wird in Abbildung 7.7 gezeigt. Das Ortslabel „Ilmenow“ wurde markiert und durch die Texterkennung wurde „Lmenow“ erkannt und in die entsprechenden Textfelder auf der rechten Seite eingetragen. In diesem

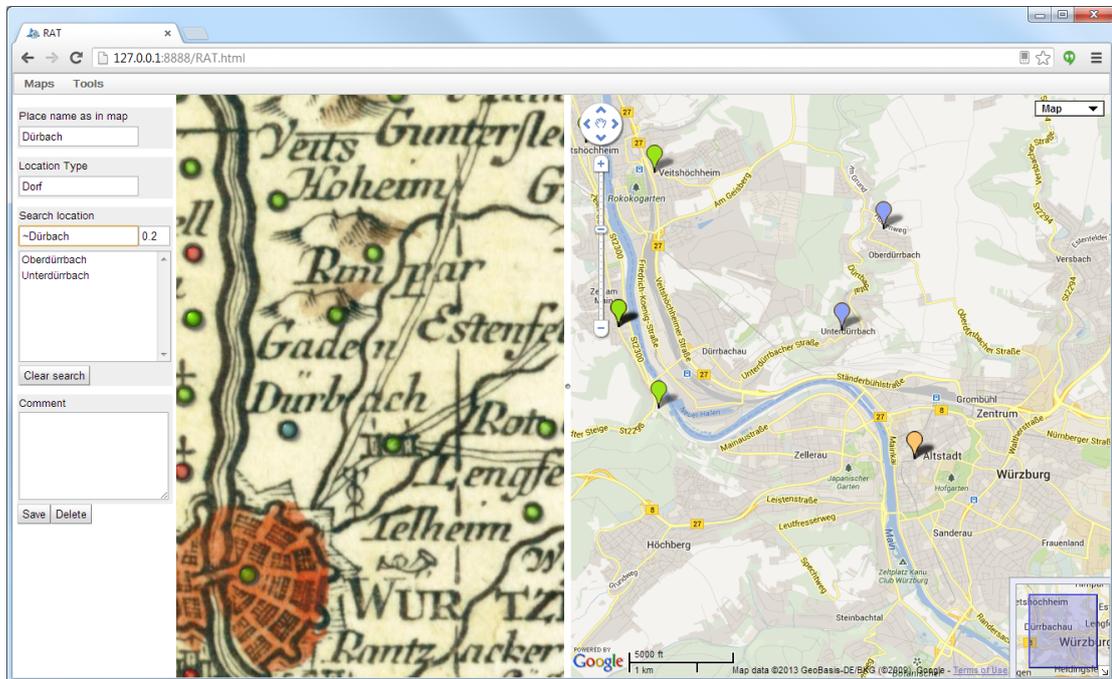


Abb. 7.5.: Nicht eindeutig zuordenbare Ortssignaturen.



Abb. 7.6.: Annotation eines Ortssignals

Fall muss der Erkennungsfehler manuell korrigiert werden. Durch die in Kapitel 5.1 vorgestellte Suche zur Unterstützung der Georeferenzierung wurde aber dennoch der korrekte moderne Ortsname als erster Treffer vorgeschlagen.

Um die Georeferenzierung von Ortssignaturen zu unterstützen wird eine Datenbank mit Orten und den zugehörigen Geokoordinaten verwendet. Das System bildet die Pixelkoordinaten aus den Altkarten mit einer „Schätzfunktion“ auf Geokoordinaten ab. Nach der Auswahl eines Ortes in der Altkarte schlägt das System moderne Orte für das Mapping vor, welche auf den Orten nahe der geschätzten Geokoordinaten basieren. Diese Vorschläge lassen sich mit einer Instant-Search filtern, welche exakte Präfixe oder phonetische Ähnlichkeit verwenden kann. Wenn der Benutzer schon den Ortsnamen mit der auf der Karte verwendeten Schreibweise eingegeben

7. RAT - Referenzierungs- und Annotations-Tool

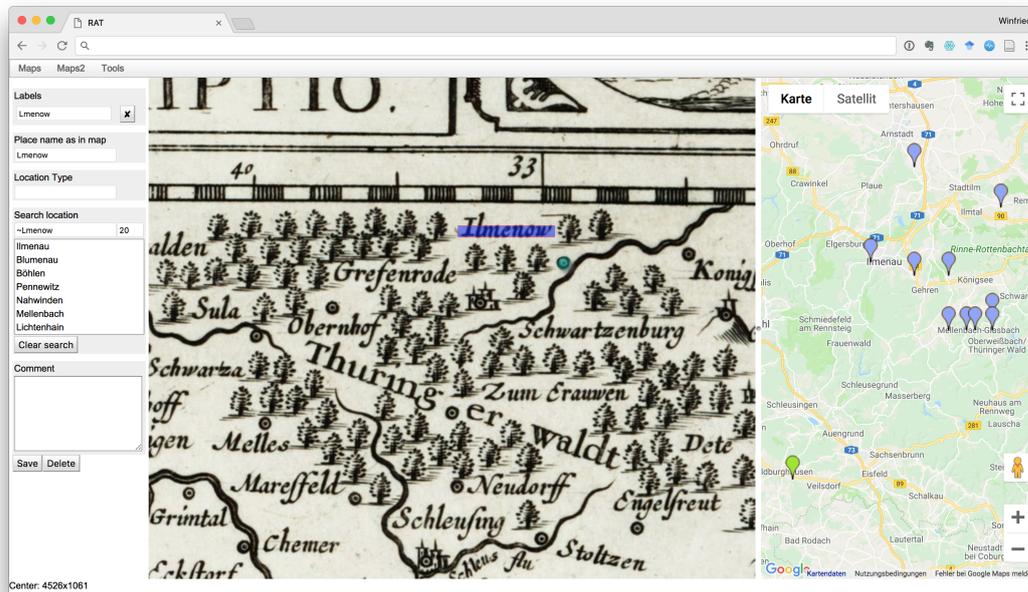


Abb. 7.7.: Für annotierte Textlabel kann eine automatische Texterkennung durchgeführt werden.

hat, dann wird dieser Name automatisch verwendet um die Ortsvorschläge nur auf die phonetisch ähnlichen einzuschränken. Die Suchergebnisse werden als Liste der Ortsnamen und durch Marker an ihren Geositionen auf der modernen Karte angezeigt. Diese zwei Anzeigen sind verknüpft um bei der Auswahl eines Ortes auch diesen Ort in der anderen Darstellung hervorzuheben.

Abbildung 7.8 zeigt das Ergebnis der Komplettreferenzierung von fast allen Orten auf der Karte. Es wurden in diesem Fall ca. 3200 Orte georeferenziert und einige Fehlen noch da sie sich nicht eindeutig zuordnen lassen.

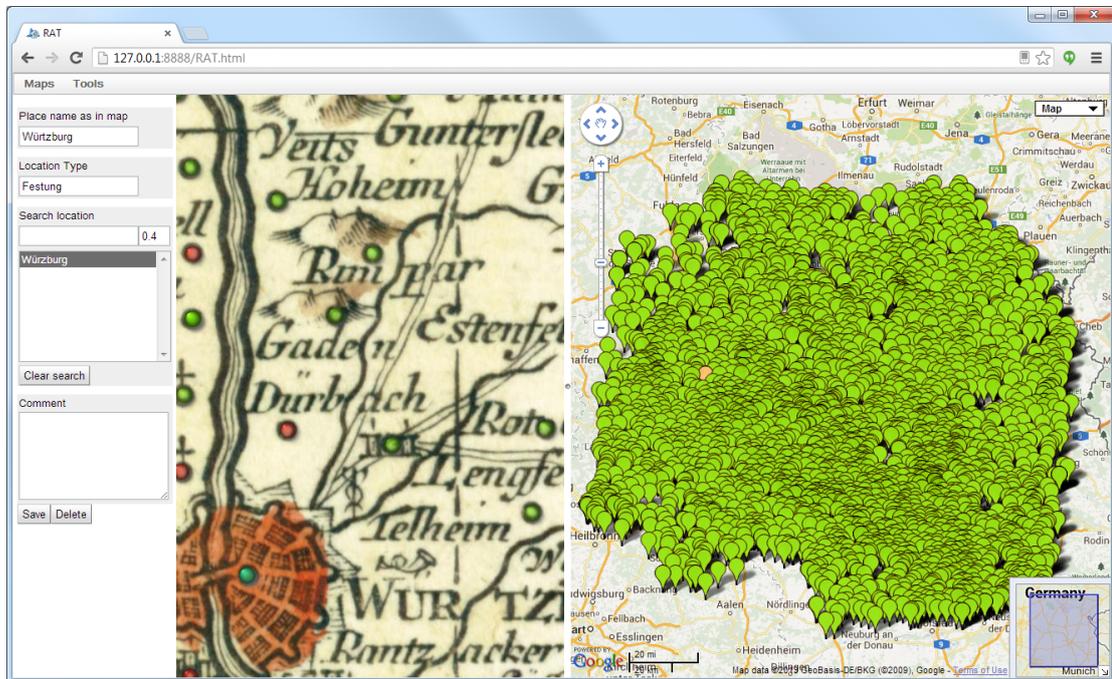


Abb. 7.8.: Durchgeführte Kompletatgeoreferenzierung.

7.3. Zusammenfassung

Es ist eine große Herausforderung und ein langer Weg, die Ergebnisse der Forschungsarbeit in einer benutzbaren Anwendungs-Software zu implementieren. Kriterien wie Skalierbarkeit, Benutzerfreundlichkeit, usw. müssen dabei zusätzlich zu den forschungsrelevanten Kriterien wie Beachtung und Verbesserung der Genauigkeit bei der Klassifikation und Extraktion der Karten-Features in den Fokus gesetzt werden.

Das Referenzierungs- und Annotationstool RAT integriert nicht nur die Resultate der Forschungsarbeit in den Bereichen automatische Extraktion und Klassifikation der Kartensymbole und der Georeferenzierung, sondern setzt auf effiziente Arbeitsabläufe. Genau diese Kombination aus Forschung und Entwicklung ist erforderlich, um den gesamten Bereich von *Digital Humanities* durch digitale Werkzeuge sinnvoll zu unterstützen. Dieses Kapitel hat einen Weg gezeigt, mit welchem Aufwand so etwas Realität werden kann.

8. Fazit

Dieses Kapitel synthetisiert die Ergebnisse dieser Forschungsarbeit, die in den Kapiteln 3 bis 7 erzielt und beschrieben wurden. Zu jeder in Abschnitt 1.2 gestellten Forschungsfrage wird zunächst eine kurze Zusammenfassung des Beitrags dieser Arbeit gegeben. Anschließend werden Schlussfolgerungen aus den dazugehörigen Erkenntnissen gezogen. Zum Schluss werden die Vor- und Nachteile dieser Forschungsarbeit im Hinblick auf die automatische Erfassung der in Altkarten enthaltenen Daten diskutiert.

Wie im Kapitel 1 motiviert, war das Ziel dieser Arbeit die Annotation von Altkarten weitgehend zu automatisieren. Speziell hat sich diese Dissertation auf die folgenden Fragen fokussiert:

1. Lassen sich Ortssignaturen in Altkarten automatisch erkennen?
2. Wie lässt sich die Beschriftung der Ortssignaturen automatisiert erfassen im Sinne von Label-Identifikation und Label-Texterkennung?
3. Wie lässt sich die Zuordnung zu modernen Orten erleichtern oder automatisieren (Georeferenzierung)?
4. Wie lässt sich die Ähnlichkeit zwischen Karten nutzen um die Erfassung der Orte zu unterstützen oder zu vereinfachen?

Um diese Forschungsfragen zu beantworten wurden vielfältige Methoden präsentiert, so dass bei der Erfassung von Ortsdaten aus Altkarten in verschiedenen Bereichen ein gewisser Grad an Automatisierung erreicht werden konnte. Es wurde gezeigt, dass sich im Bereich der Annotation von Orten in Altkarten viele der zu bewältigenden Teilaufgaben zu einem hohen Grad automatisieren lassen oder sich zumindest Vorschläge generieren lassen, um den manuellen Aufwand wesentlich zu verringern.

8. Fazit

Wie in Kapitel 3 gezeigt wurde, lässt sich die Erkennung von Ortssignaturen auf verschiedene Arten semiautomatisch durchführen. Eine Template-Matching-basierte Variante wurde mit einer auf CNNs basierenden Deep-Learning-Variante verglichen. Bei dem Template-Matching-Ansatz konnten durch die eingeführte automatische Generierung von Masken deutliche Steigerungen des F1-Score im Vergleich zum unmodifizierten Template-Matching erreicht werden. Auch in den Werten für *Precision* und *Recall* spiegelt sich dies wieder, und sie liegen entweder auf dem gleichen Niveau oder das modifizierte Template-Matching übertrifft das Template-Matching um mehr als das Zehnfache. Beim Vergleich des modifizierten Template-Matching mit dem CNN-Ansatz ergibt sich kein so eindeutiges Bild, und je nach Karte unterscheidet sich, welches Verfahren den besseren F1-Score erreichen kann. Bei den Vergleichen mittels *Precision-Recall*-Kurve lässt sich sehen, dass der Template-Matching-basierte Ansatz für hohe *Precision*-Werte einen besseren *Recall* liefert und der CNN-basierte Ansatz für niedrigere *Precision*-Werte den besseren *Recall* liefert. Je nachdem wo sich nun die *Precision-Recall*-Kurven überschneiden führt dies für ein anderes Verfahren zu einem besseren F1-Score. Welches der Verfahren besser geeignet ist, hängt dadurch auch von der weiteren Verwendung der Daten ab, und ob dabei mehr Wert auf eine hohe *Precision* oder einen hohen *Recall* gelegt wird.

Aus diesen Erkenntnissen ergeben sich die folgenden theoretischen Implikationen:

1. Diese Methoden können in anderen Domänen der Objekterkennung in Graphiken verwendet werden, um standardisierte oder zumindest sich wiederholende Symbole zu finden. Dennoch muss man verstehen, dass um so weniger die Symbole standardisiert sind, um so niedriger die Erkennungsraten ausfallen können.
2. Die künstlerischen Freiheiten, die Kartenmacher damals hatten, machen eine manuelle Nachbearbeitung der automatischen Symbolerkennung unvermeidlich, denn ein kleiner aber wichtiger Teil der Ortssignaturen in Altkarten wurde mit Absicht damals anders dargestellt.
3. Diese Methoden lassen sich nicht auf die Bereiche übertragen, in denen es um die Erkennung variabler Formen geht, wie beispielsweise Gesichtserkennung.

Die Ergebnisse aus Kapitel 4 zeigen, dass für die Label-Identifikation das angepasste klassische Verfahren zur Trennung von Text und Grafik einer Deep-

Learning-basierten Identifikation der Textbereiche deutlich unterlegen ist, auch wenn das Deep-Learning-Verfahren auf einem Outdoor-Foto-Datensatz trainiert wurde. Durch ein Feintuning auf Kartendaten konnte nochmals eine deutliche Steigerung der Erkennungsrate erreicht werden, wobei auch eine genauere Übereinstimmung der Bounding-Boxen erzielt wurde. Die bessere Übereinstimmung der Bounding-Boxen ist auf das Training mit konsistent an Grund- und x-Höhe ausgerichteten Bounding-Boxen zurückzuführen.

Die Erkenntnisse dieser Arbeit lassen sich auf die Gesamtdomäne der Karten und die Bereiche der Text/Graphic Separation und Scene-Text Detection übertragen, und folgendes lässt sich schließen:

1. Die modernen Karten sind standardisiert und computergeneriert, die Ungenauigkeiten, die für Altkarten typisch sind, lassen sich in modernen Karten nicht finden, deswegen würde man in modernen Karten bereits mit regelbasierten Verfahren einen hohen Genauigkeitsgrad bei der Textlabelerkennung erreichen. Dennoch liefern CNNs ein zur Zeit unschlagbares Werkzeug, um Textlabels in Karten jeglicher Art zu erkennen.
2. Die Regeln, die für Text-Graphic-Separation aufgestellt wurden, um textuelle Symbole von nicht-textuellen Symbolen zu unterscheiden, die in die gleiche Größenordnung fallen, sind domänenunabhängig und können auch für andere ähnliche Datensätze verwendet werden.
3. Dagegen müssen die statistischen Verfahren für jeden neuen Anwendungsfall, der sich signifikant von den bisherigen Trainingsdaten unterscheidet, neu trainiert werden.

Des Weiteren wurde in Kapitel 4 auch die Texterkennung von manuell ausgeschnittenen Labels mit Hilfe von Deep-Learning-Verfahren untersucht. Hierbei wurde der Schwerpunkt darauf gelegt, dass ein effizientes Training mit einer kleinen Datenmenge möglich ist. Dies war erforderlich, da die Karten nur eine begrenzte Anzahl an Ortslabels enthalten, meist im Bereich von mehreren hundert bis zu einigen tausend. Dadurch würden Verfahren, die mehrere tausend oder noch einige Größenordnungen mehr an Trainingsbeispielen benötigen, wie es bei vielen Deep-Learning-Systemen typisch ist, nicht genug Trainingsdaten haben, um überhaupt

8. Fazit

trainiert werden zu können. Im Rahmen dieser Dissertation konnte ein Modell gefunden werden, welches mit wenigen hundert Beispielen trainiert werden kann und dennoch sehr geringe Zeichenfehlerraten erreicht. Die Texterkennung, die mit wenigen hundert Trainingsbeispielen auskommt, hat allerdings ihren Preis. Die Texte müssen dafür genau ausgerichtet werden, das heißt die Grundlinie und die x-Linie der ausgeschnittenen Textlabels müssen an festgelegten Positionen liegen.

Diese Erkenntnis hat folgende theoretische Implikationen für den OCR-Bereich insgesamt:

1. Deep-Learning-Verfahren lassen sich für OCR-Aufgaben mit deutlich weniger Trainingsdaten durchführen, wenn eine Textausrichtung als Vorverarbeitungsschritt stattfindet. Dies ist speziell hilfreich in Bereichen, in denen nur wenige Trainingsdaten zur Verfügung stehen, wie historische Schriften.
2. Die alten Drucke, die im Druckverfahren aus beweglichen Lettern entstanden, fallen in eine ähnliche Domäne wie Altkarten, denn jeder der Beweglichen Lettern war ein Unikat. Die Lettern, die denselben Buchstaben repräsentieren, sind also ähnlich, aber nicht identisch. Man könnte zwar ausreichend Trainingsdaten erstellen, indem ein Großteil des Druckes abgetippt wird, aber mit weniger Trainingsdaten würde man die Handarbeit minimieren. Wenn die Layout-Analyse die Zeilen als gut genug ausgerichtet erkennt, könnte man mit weniger Trainingsdaten auskommen.

Die mögliche Unterstützung durch Generierung von Vorschlägen wurde in Kapitel 5 behandelt. Hierbei konnte an einer Beispielparte gezeigt werden, dass unter den Top 10 der Zuordnungsvorschläge mit 88,9% der richtige Ort enthalten ist für Orte die vorher manuell zugeordnet werden konnten, wenn die Position auf 0, 25° Breite und Länge vorhergesagt werden kann. Bei Betrachtung der Top 15 Vorschläge ist in über 90% der richtige Ort enthalten. Hierdurch ist eine schnelle Zuordnung ohne manuelle Suche möglich. Als Fazit im Bereich Georeferenzierung ergibt es sich aus den Erkenntnissen dieser Arbeit:

1. Die Lösung ist kartenspezifisch und für moderne Karten uninteressant, denn für ist nach der Georeferenzierung weniger Punkte eine Umrechnung von Pixel- in Geokoordinaten möglich.

2. Nicht kartenspezifisch ist das Matching der Ortsnamen. Das hier beschriebene Verfahren kann überall verwendet werden, wo man den Ortsnamen in der historischen Schreibweise finden kann, z. B. alte Texte oder topographische Lexika, um eine Vorschlagsliste von möglichen modernen Ortsnamen zu generieren.

In Kapitel 6 wurde gezeigt wie zu einer Karte ähnliche Karten, welche annähernd die selben Ortsdaten mit ähnlichen Verzerrungen enthalten, gefunden werden können und eine Zuordnung zwischen den Ortssignaturen der Karten hergestellt werden kann. Es wurde auch gezeigt, dass diese Zuordnung über einen weiten Bereich von *Precision* und *Recall*, bei der Erkennung der Ortssignaturen, stabil funktioniert. Diese Zuordnung hat vielfältige Anwendungsmöglichkeiten, z. B. die Erkennung von Fehlern in einer automatischen Annotation, die Auflistung von Unterschieden zwischen Karten oder die Zuordnungen zwischen Karten innerhalb derer voneinander kopiert wurde oder um eine Genauigkeitsanalyse zwischen den Karten durchzuführen.

Aus den Erkenntnissen dieser Forschungsarbeit in diesem Bereich ergeben sich folgende theoretische Implikationen:

1. Wenn man bereits eine Datenbank annotierter Altkarten hat, ist es mit diesen Techniken einfach, eine neue Altkarte, die Ähnlichkeiten zu einer bereits vorhandenen Karte aufweist, zu annotieren. Hierzu müssen nur wenige Orte auf der neuen Karte georeferenziert werden. Somit wäre es wichtig, einen offenen Zugang zu solchen Kartensammlungen zu ermöglichen.
2. Eventuell existierende Kartendokumente in verschiedenen Sprachen über die gleiche geographische Region können somit einfacher gefunden und annotiert werden. Die Textlabels in verschiedenen Sprachen für die gleiche Region können so automatisiert gewonnen werden.

Die Implementierung wird in Kapitel 7 gezeigt, welche zum Erfassen von Trainings- und Testdaten verwendet werden kann und auch genutzt werden kann um die Automatisierungen, welche in den vorherigen Kapiteln 3 bis 6 beschrieben wurden, zu testen sowie automatisch erfasste Daten zu korrigieren.

8. Fazit

Die Erkenntnisse über das Kartenmaterial an sich, die durch die Ergebnisse dieser Arbeit gewonnen werden können, haben Auswirkungen auf den Wissenstand in historisch-geographischen Untersuchungen über die Geschichte der Orte und Geschichte der Kartendokumente. Das zeigt noch einmal, wie groß das Potential solcher digitalen Werkzeuge im ganzen Bereich *Digital Humanities* ist.

Die in dieser Forschungsarbeit vorgestellten Methoden haben die folgenden Vorteile:

- Automatisierungen, die auf Verfahren des maschinellen Lernens basieren, machen die Anwendung neuer Techniken einfach, wenn ein passendes Trainingsset zur Verfügung steht.
- Die OCR-Verfahren könne mit mehr Trainingsdaten, von verschiedenen Karten, auch auf unbekanntem Karten eine hohe Erkennungsqualität bieten.
- Der Regelbasierte Teil der Text-Graphic-Separation ist ohne Anpassung auf andere Karten übertragbar.
- Der Aufwand bei manueller Arbeit, die für die Erstellung der Trainingsdaten benötigt werden, ist geringer, als eine komplette manuelle Annotation.
- Man braucht keine besonders ausgebildeten Experten, um mit den getesteten Techniken in der Altkartendomäne zu arbeiten.
- Die verschiedenen Automatisierungen sind unabhängig von einander aufgebaut, sodass sie einzeln oder gemeinsam nutzbar sind.

Trotz aller Vorteile, die folgenden Schwachpunkte lassen sich feststellen:

- Keine Integration in andere Systeme wurde getestet.
- Für jede Karte ist manuelle Arbeit nötig, um Trainingsdaten zu generieren.
- Die Fehlerrate ist zwar gering, aber dennoch vorhanden.
- In manchen Bereichen ist es unmöglich, genug Trainingsdaten zu erfassen, zum Beispiel bei einer geringen Anzahl von Symbolen eines Typs.

Um die Schwächen der Methoden und der Technologie auszugleichen, sind folgende Wege möglich. Zum einen können Trainingsdaten aus den Benutzerannotationen anderer Software zur Kartenannotation generiert werden. Die Nutzer der Softwa-

re könnten motiviert werden, die Karten semi-automatisch zu annotieren, weil der Annotationsaufwand deutlich reduziert wird. Dank der Unabhängigkeit der Komponenten, die in dieser Arbeit vorgestellt wurden, sind Einzelteile davon leicht in verschiedene Systeme integrierbar. Darüberhinaus bietet die bisher nicht vorhandene Integration in andere Systeme die Möglichkeit die Schnittstellen zum Datenaustausch noch frei zu definieren.

Bereits existierende Systeme zur manuellen Kartenannotation bieten die Möglichkeit die Automatisierungstechniken in diese zu integrieren und somit deren vorhandene Schnittstellen zum Datenaustausch zu nutzen. Bereits existierende Systeme zur manuellen Kartenannotation bieten die Möglichkeit die dort bereits anfallenden Annotationsdaten als Trainingsdaten zu nutzen und mit mehr Daten auch Modelle zu trainieren die über verschiedene Karten hinweg funktionieren. Für ähnliche Domänen wird möglicherweise auch an einer Reduktion der zum Training benötigten Daten gearbeitet und dies kann in die Altkartendomäne übertragen werden.

9. Ausblick

Durch die Vielzahl an existierenden Techniken, die in die Domäne der Altkarten übertragen werden können, um zur Erkennung der Ortssignaturen und des Textes genutzt zu werden, ist eine weitere Forschung auf diesem Gebiet interessant. Auch durch immer wieder neu eingeführte Techniken ergibt sich Verbesserungspotential, allerdings ist der Fortschritt auf dem Gebiet der Texterkennung und Objekterkennung, welches Teilbereiche sind deren Ergebnisse hier genutzt werden können, so rasant, dass es schwierig erscheint, mit den neuen Forschungsergebnissen Schritt halten zu können.

Ein Ansatz diese Forschungsaktivität zu nutzen könnte darin bestehen, einen annotierten Datensatz zu erstellen um einen Wettbewerb, vergleichbar zur *ICDAR Robust Reading Challenge* (Karatzas et al., 2015), durchzuführen. Dadurch können interessierte Forscher ihre neuen Methoden direkt an diesen Daten testen.

Auch die Ausnutzung der Synergieeffekte sollte in diesem Wettbewerb ein Bestandteil sein. Idealerweise würde dies aus einer Plattform bestehen die für eine detaillierte manuelle Annotation ausgelegt ist, zusätzlich aber entsprechende Schnittstellen bietet, um diese Annotationen durch externe Systeme erstellen zu lassen und diese einzufügen. Dadurch können für alle Automatisierungen Baseline-Implementierungen bereitgestellt werden mit deren Hilfe es dann möglich ist einzelne neue Ansätze auch im Zusammenspiel mit den anderen Modulen und den von diesen bereitgestellten Informationen zu testen. Zudem kann auch das Zusammenspiel mehrerer neuer Ansätze getestet werden. Darüber hinaus würden die neuen Ansätze durch dieses Vorgehen einheitliche Schnittstellen unterstützen und könnten damit leicht in den Praxiseinsatz gebracht werden, auch in verschiedenen Systemen, die die entsprechende Schnittstelle implementieren.

9. Ausblick

Besonders interessant für weitere Untersuchungen erscheint das Verknüpfen der Informationen aus den verschiedenen Bereichen, um damit die Erkennungsgenauigkeit zu verbessern. So kann die Vorhersage der ungefähren Lage des Ortes genutzt werden um eine Liste in Frage kommender Orte zu erstellen, diese Liste kann dann verwendet werden um für das OCR eine Einschränkung der zu erwartenden Namen vornehmen zu können. Auch kann wiederum das OCR Ergebnis eines Labels benutzt werden um dessen Zuordnung zu einer Ortssignatur zu erleichtern. Zusätzlich bietet die Identifikation von ähnlichen Karten ein großes weiteres Automatisierungspotential durch die Verknüpfung der Informationen aus den Altkarten.

Eine weitere mögliche zukünftige Aufgabe ist die automatische Umwandlung von Altkarten in Vektordaten, dies beinhaltet die Erfassung von Punkt-, Linien- und Polygon-Features. Die Einteilung in diese drei Feature-Arten wird auch in GIS-Software verwendet. Es können dann verschiedene Ebenen definiert werden, die jeweils genau einen dieser Feature-Typen enthalten. Dabei können zu jedem Feature noch zusätzliche Metainformationen abgespeichert werden. So könnte eine Ebene definiert werden, die Orte als Punktfeatures beinhaltet. Solche Angaben wie der Ortsname und die Einwohnerzahl können als Zusatzinformationen zu jedem Punkt abgelegt werden. Zum abspeichern von Flussläufen könnte eine Ebene mit Linienfeatures verwendet werden. Für Wälder und Gebiete, z.B. Länder oder Territorien, können wiederum Polygon-Features verwendet werden. Durch die mehrfache Verwendung von Feature-Typen für verschiedene Ebenen, welche verschiedene Objekte der realen Welt abbilden, kann sich auch die Darstellung dieser Ebenen in der Karte erheblich unterscheiden. So werden z. B. Waldflächen oft durch eine Ansammlung von Bäumen symbolisiert und Gebiete durch ihre Grenzverläufe, die dann häufig als gestrichelte oder gepunktete Linien dargestellt sind. Zusätzlich sind noch Annotationen und Verzierungen zu erfassen, welche im Gegensatz zu den oben beschriebenen Features keine geographische Entsprechung in der realen Welt haben.

Dadurch dass die Karten ohne ein definiertes Regelwerk entstanden, das vorschreiben würde, was erlaubt ist und was nicht und welche Daten in welcher Art dargestellt werden, und auch künstlerische Aspekte in die Erstellung der Altkarten eingingen, wird ist eine vollständige automatische Erfassung der in Altkarten enthaltenen Daten insgesamt als unrealistisch anzusehen. Es wird immer wieder

Karten geben die Konventionen in einzigartiger Weise ignorieren und so auch für Menschen nicht eindeutig interpretierbar sind oder zusätzliches externes Wissen und die Fähigkeit zum Schlussfolgern benötigen um verstanden zu werden. Dieses Manko kann auch nicht durch ein mehr an Trainingsdaten ausgeglichen werden. Andererseits sind Karten im Allgemeinen darauf ausgelegt, zumindest in den Hauptmerkmalen, ohne Anleitung verstanden zu werden. Somit ist nicht zu erwarten, dass dies eine weitreichende Beeinträchtigung darstellen würde. Außerdem kann, falls nur in einigen Fällen z.B. für einen speziellen Ort, eine Abweichung von der allgemein verständlichen Systematik besteht, die Erkennung darauf ausgelegt werden, unklassifizierbare Bestandteile einer Karte als manuell zu annotierende Spezialfälle zu kennzeichnen.

Ein weiteres Feld sind Forschungsmöglichkeiten, die sich erst durch die fortschrittliche Automatisierung und hierdurch erfassbaren Daten ergeben. Z. B. für die Untersuchung von Kartenzuständen, in denen kleine Änderungen von vielleicht nur einem neu hinzugefügten Ort unter tausend auf der Karte vorhandenen Orten entdeckt werden müssen. Hierzu müssen die Orte auf mehreren Karten mit sehr hoher Genauigkeit erfasst werden und dann einander zugeordnet werden. Dabei unterscheiden sich die Koordinaten innerhalb jeder einzelnen Karte, auch wenn sie von derselben Kupferplatte gedruckt wurden, da die Karten verschieden eingescannt wurden. Zudem ergeben sich auch durch die Alterung leichte Verzerrungen über die Kartenfläche. Ein weiterer Unterschied, der ausgeblendet werden muss, entsteht durch die nachträgliche Handkolorierung der Karten.

Anhang A.

Karten

Die in dieser Arbeit verwendeten Karten sind in zwei geographische Regionen aufgeteilt zum einen Karten die den ehemaligen Fränkischen Reichskreis abdecken, zum anderen Karten die Luxemburg beinhalten. Die Karten stammen von Franconica online, der David-Rumsey Map Library und der Nationalbibliothek Luxemburg und decken eine Zeitraum vom 16. bis zum 19. Jahrhundert ab.

A.1. Luxemburg

A.1.1. Lutzenburgensis Ducatus / Pieter van den Keere (1617)

Eine Luxemburg-Karte von Pieter van den Keere, beschrieben auf David Rumsey Map Collection (Rumsey Historical Map Collection, 2018) als:

First edition, and the first complete atlas of an entire country. Includes the Leo Belgicus plate.



Abb. A.1.: Pieter van den Keere (1617)

Petri Kaerii Germania Inferior id est, XVII Provinciarum ejus novae etc exactae Tabulae Geographicae, cum Luculentis Singularum descriptionibus additis. a Petro Montano.

Amstelodami impensis Pet. Kaerii. 1617.

Kupferstich, 38 x 50 cm

Signatur: David Rumsey Historical Map Collection 12070.031.

A.1.2. Lutzenburgensis Ducatus / Johannes Janssonius van Waesberge (1636)

Eine Luxemburg-Karte von Johannes Janssonius van Waesberge, beschrieben auf David Rumsey Map Collection (Rumsey Historical Map Collection, 2018) als:

Copper engraved map of Namur Province, Belgium. Shows towns, villages, landmarks, mountains, forests and rivers. Relief shown pictorially.



Abb. A.2.: Johannes Janssonius van Waesberge (1636)
 Lutzenburgensis Ducatus. Jacobo. Surhonio Montensi Auctore. (to accompany) Atlas sive Cosmographicae Meditationes de Fabrica mundi et fabricati figura. De novo multis in locis emendatus novisque tabulis auctus. Amsterodami, apud Iohannem Ianssonium van Waesberge. Anno, 1673.
 David Rumsey Historical Map Collection: 11487.067.

First reworked edition of Mercator's Atlas Minor issued by the Janssonius heirs (Johannes Janssonius van Waesberge) in 1673, without text or printed index. With engraved architectural and emblematic title page, and 183 copperplate engraved maps. Printed from the plates of the Cloppenburg editions of the Atlas Minor (1630, 1632, 1636), with a new map of the United Provinces and additional decorative elements to some maps, along with some updates to the maps. This edition of Mercator's 'Atlas'... was a changed format edition. Its newly engraved maps fell in between the folio maps in the Mercator-Hondius Atlas and the Atlas Minor. The Cloppenburg edition seems to have been suppressed after 1636. It was continued with this copy after 1673 by the Janssonius heirs. Because the maps are all take from the Cloppenburg

Anhang A. Karten

plates of the 1636 edition, these maps are dated 1636, while the atlas is dated (Pub Date) as 1673. The maps include decorative cartouches, latitudinal and longitudinal lines, coats of arms, compass rose, and show political boundaries, cities, towns, topography, vegetation, waterways and landmarks. Relief shown pictorially.

A.1.3. Carte du Duché de Luxembourg / Didier Robert de Vaugondy (1753)



Abb. A.3.: Didier Robert de Vaugondy (1753)
Carte du Duché de Luxembourg, dans laquelle on trouve la partie Meridionale & limitrophe de la Seigneurie de Lyege. Par le Sr. Robert de Vaugondy Geographe Ord. du Roy. Avec privilege 1753. E. Haussard sculp.
Signatur: David Rumsey Historical Map Collection 3353.052.

Eine Luxemburg-Karte von Didier Robert de Vaugondy, beschrieben auf David Rumsey Map Collection (Rumsey Historical Map Collection, 2018) als:

1st edition, with five postal maps added, as issued. With the list of

subscribers and the extra postal maps of England, France, Germany, Italy and Spain. The French postal map is dated 1758. All five were added after the printed list of maps was printed, but are mentioned by Vaugondy in the preface, so they may have been an afterthought. An owner has added their titles to the printed list of maps, in ms. Engraved title page, Advertisement, and Geographical essay giving the sources of the maps. This atlas was reissued until 1799 (Phillips), with later issues having a map of the United States. In original half leather patterned paper covered boards with spine reading „Nouvel Atlas De Mrs. Robert.“ One of the first atlases based on scientific surveys, in the tradition of D’Anville and De L’Isle. Outline color.

A.1.4. Lutzembvrgensis / Abraham Ortelius (1608)

Eine Luxemburg-Karte von Abraham Ortelius, beschrieben auf David Rumsey Map Collection (Rumsey Historical Map Collection, 2018) als:

Koeman: “This the first edition ever printed with Italian text. The heirs of Abraham Ortelius sold a manuscript translation by Filippo Pigafetta to Vrients, which was then printed by him. In this translation, the Parergon and the Nomenclator were included. Apart from this, Vrients also added at the end, an Introduction to Cosmography, written by Michel Coignet. Several new maps appear here for the first time.” Vrients added 8 new maps, making this combined Theatrum and Parergon edition the largest with 166 maps. 128 maps in the “Theatrum”, with the “Parergon” having 38 maps, a portrait of Pope Clement VIII., dedications to Pope Clement VIII. and to Cardinal Aldobrandino, 1605. The Theatrum and Parergon plates are uncolored. The Parergon is Ortelius’ atlas of ancient geography.

Koeman: “This atlas of ancient geography must be regarded as a personal work of Ortelius. For this work he did not, as in the Theatrum, copy other people’s maps but drew the originals himself which were later engraved by Jan Wierix i.a. He took many places and regions

Anhang A. Karten



Abb. A.4.: Abraham Ortelius (1608)

Theatro Del Mondo Di Abrahamo Ortelio: Da lui poco inanzi la sua morte riveduto, & di tavole nuove, et commenti adorno, & arricchito con la vita dell'Autore. Traslato in Lingua Toscana dal Sigr. Filippo Pigafetta. In Anversa, Appresso Giovanni Bapta. Vrintio, M.DC.VIII. (with) Parergon, Cioe Fuor D'Opera, Et Giusta, Overo Alcune Tavole Dell'Antica Geographia. (with) Nomenclator Ptolemaicus; Omnia Locorum Vocabula Quae In Tota Ptolemaei Geographia.

Signatur: David Rumsey Historical Map Collection 10001.143.

from the lands of classical civilization to illustrate and clarify their history, a subject very close to his heart... The maps and plates of the Parergon have to be evaluated as the most outstanding engravings depicting the wide-spread interest in classical geography in the 16th century.”

A.1.5. Lutzembvrgensis Ducatus Veriss. Descript. / Abraham Ortelius (1579–1612)

Eine Luxemburg-Karte von Abraham Ortelius, von welcher verschiedene Kartenzustände in der Nationalbibliothek vorhanden sind. Die Karten sind Abdrucke der selben Kupferplatte wie *Lutzembvrgensis / Abraham Ortelius (1608)* A.1.4 und unterscheiden sich nur in der Koloration und an der Kupferplatte vorgenommenen Korrekturen.

Die Karten werden in Fritzen und Solchenbach (2017) folgendermaßen beschrieben:

Das *Theatrum Orbis Terrarum* von Abraham Ortelius (1527-1598) gilt als der erste moderne Atlas, der von 1570 bis 1612 in 31 Auflagen und 7 Sprachen herausgegeben wurde. Der Atlas erschien zunächst mit 70 Karten und wuchs in seinen letzten Ausgaben auf bis zu 167 Karten an, dazu existiert eine umfangreiche Fachliteratur. Seit 1579 enthält das *Theatrum* auch eine Karte des damaligen Herzogtums Luxemburg (*Lutzenburgensis Ducatus veriss[ima] Descript[io]*, also Wahre Beschreibung des Herzogtums Luxemburg), die von Jakob Surhon schon 1550 erstellt wurde, aber lange Jahre nicht erscheinen durfte. Die Luxemburg-Karte war seit der lateinischen Ausgabe von 1579 Bestandteil des *Theatrum*. Allerdings gab es im Lauf der Zeit verschiedene Zustände der Luxemburg-Karte. Zunächst wurde der kartographische Inhalt reichhaltiger, später gab es Korrekturen, z.B. in den Schraffuren der Kartusche. Bisher waren fünf verschiedene Zustände bekannt, im vorliegenden Beitrag werden erstmals sieben Zustände beschrieben und dokumentiert, die sich (bis auf den letzten Zustand) im Bestand der *Réserve précieuse* der *Bibliothèque Nationale de Luxembourg* befinden.

Die Luxemburg-Karte im Ortelius-Atlas *Theatrum Orbis Terrarum* ist eine der ersten Karten des Herzogtums im 16. Jahrhundert. Der erste Zustand geht auf Jakob Surhon zurück. Die zusätzlichen kartographischen Inhalte in den Zuständen 2-4 stammen von Jan van Schille und weiteren zeitgenössischen Kartographen (vermutlich Till-

mann Stella, Christian Sgrooten und Frans Hogenberg). Die Zustände 5-7 unterscheiden sich nur noch in den Gravuren.

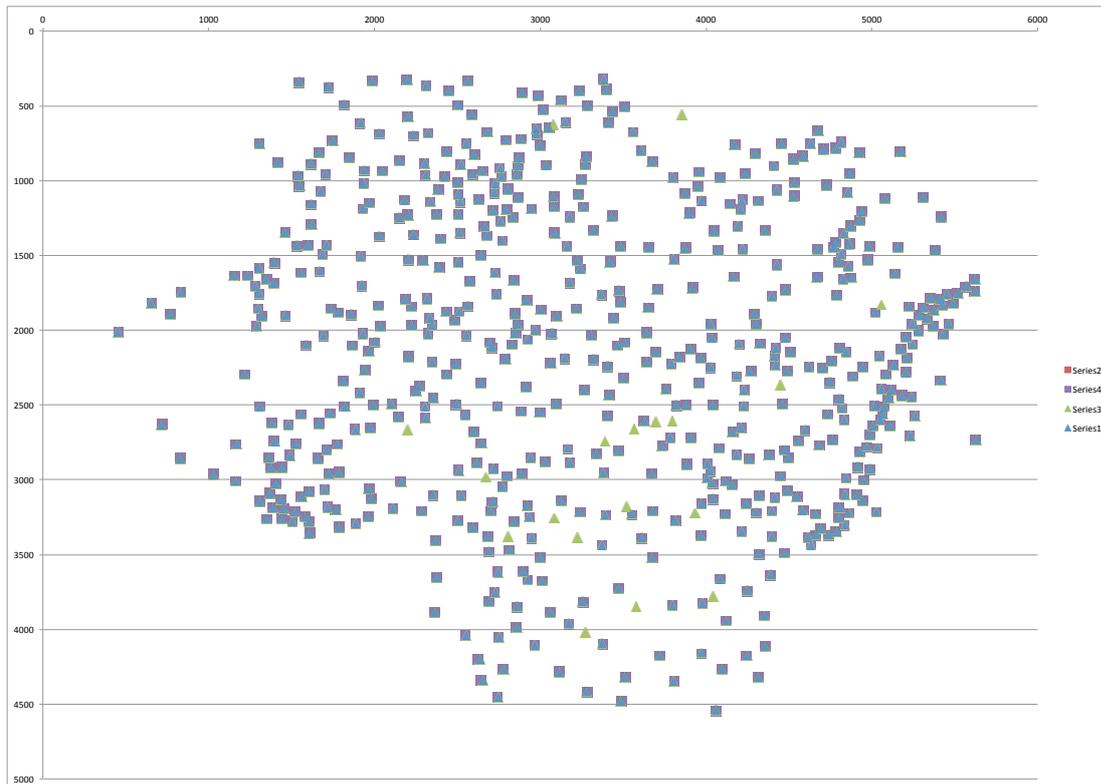


Abb. A.5.: Darstellung der Überlagerung der Orte aus den Zuständen Z2 und Z3 der Ortelius 1579 (A.1.5) Karte. Es wurden 18 zusätzliche Ortssignaturen gefunden (grün) und keine Ortssignatur wurde entfernt (rot).

A.2. Fränkischer Reichskreis

A.2. Fränkischer Reichskreis

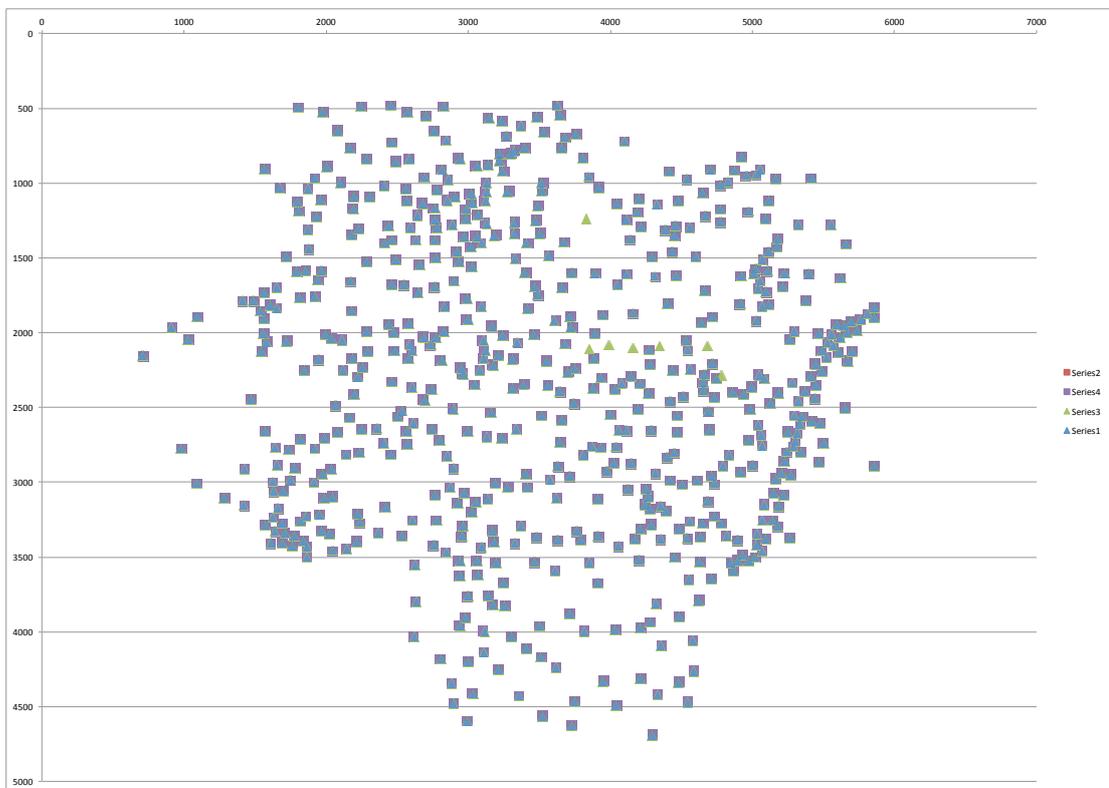


Abb. A.6.: Darstellung der Überlagerung der Orte aus den Zuständen Z3 und Z4 der Ortelius 1579 (A.1.5) Karte. Es wurden sieben zusätzliche Ortssignaturen gefunden (grün) und keine Ortssignatur wurde entfernt (rot).

Anhang A. Karten

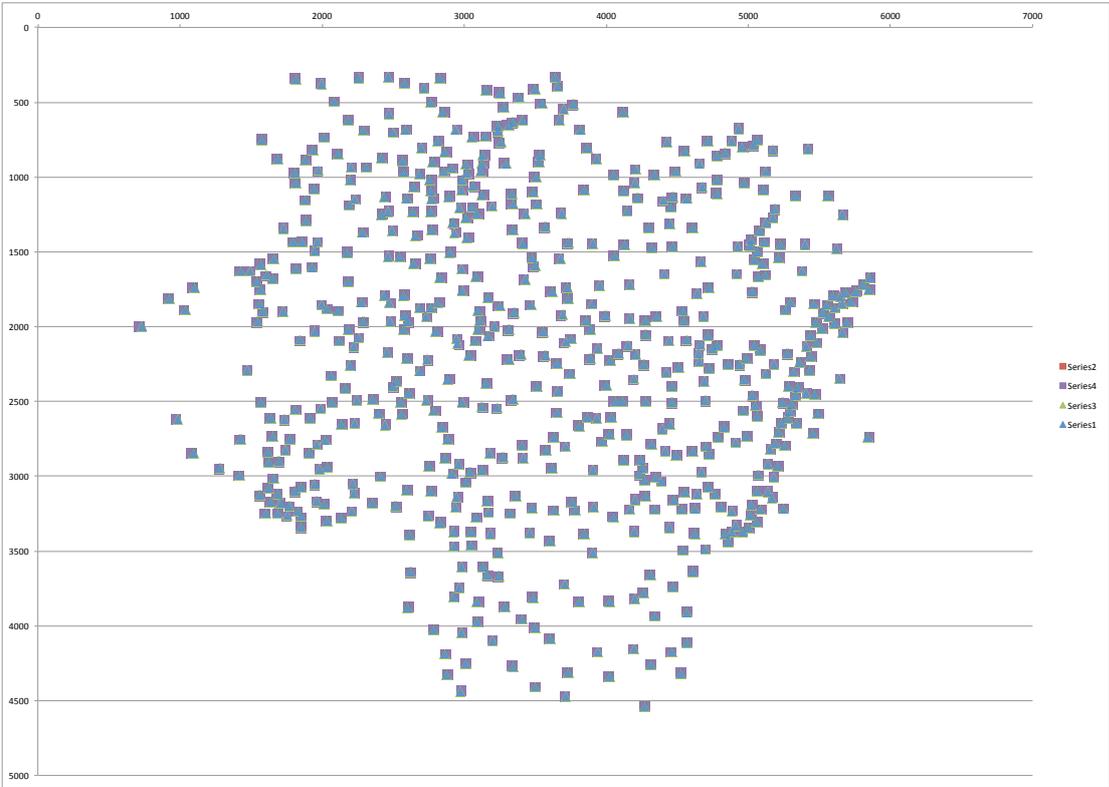


Abb. A.7.: Darstellung der Überlagerung der Orte aus den Zuständen Z4 und Z5 der Ortelius 1579 (A.1.5) Karte. Es wurden keine Unterschiede in den Ortssignaturen gefunden.

A.2.1. Das FranckenLandt = Chorographi Franciae Orien[talis] / Sebastian von Rotenhan (1533)



Abb. A.8.: Sebastian von Rotenhan (1533)
Das FranckenLandt = Chorographi Franciae Orien[talis] / Sebastian von Rotenhan. - Auff new widerumb in Truck. - München: Peter König, ca. 1620.
- Holzschnitt und Kupferstich, 56,4 x 59,6 cm.
Signatur: Universitätsbibliothek Würzburg 36/G.f.m.9-14,136.

Eine Karte des Fränkischen Reichskreises, beschrieben auf Franconica-Online

Anhang A. Karten

(Franconica-Online Historische Karten, 2018) als:

Sebastian von Rotenhan (1478-1532) war Hofmeister des Fürstbischofs von Würzburg, Konrad von Thüngen. Im Bauernkrieg verteidigte er die Feste Marienberg. Als guter Kenner seiner Heimat entwarf er, wohl einem Aufruf Sebastian Münsters von 1528 folgend, die erste Landtafel Frankens. Sie wurde erst nach seinem Tod 1533 von Petrus Apian in Ingolstadt zum ersten Mal gedruckt und erlebte viele Nachstiche.

Noch um 1620 druckte der Münchner Kunsthändler Peter König von den Originalholzstöcken die Karte nach. Er fügte jedoch in Kupferstich eine neue Legende am unteren Rand hinzu und widmete die Karte dem damaligen Fürstbischof von Würzburg und Bamberg, Johann Gottfried von Aschhausen. Dessen Wappen steht im Mittelfeld der Legende. Die Karte selbst ist nach Süden ausgerichtet ('Mittag' steht am oberen Rand, 'Mittnacht' am unteren, 'Aufgang' als Osten links, 'Niedergang' rechts), steht also nach unseren heutigen Vorstellungen auf dem Kopf. Sie umfasst das Gebiet zwischen Wetterau und dem Vogtland, den Ausläufern der Fränkischen Alb und dem Thüringer Wald. Meilenmessstab und Zirkel geben ein Bild der Entfernungen. Für die Orte ist eine Zeichenerklärung vorhanden. Die Karte ist das bisher einzige bekannte Exemplar. Sie stammt aus einem Kartensammelband des Zisterzienserklosters Ebrach, der durch die Säkularisation an die Universitätsbibliothek gelangte.

A.2.2. Franciae orientalis (vulgo Franckenlant) descriptio / Sebastian von Rotenhan/Abraham Ortelius (nach 1570)

Eine Karte des Fränkischen Reichskreises, beschrieben auf Franconica-Online (Franconica-Online Historische Karten, 2018) als:

Der Niederländer Abraham Ortelius (eigentlich Oertel, 1527-98) brachte 1570 den ersten 'modernen' Atlas, 'Theatrum orbis terrarum', heraus, der auf Vorlagen verschiedener Kartographen zurückging. Dieser erlebte

A.2. Fränkischer Reichskreis



Abb. A.9.: Sebastian von Rotenhan/Abraham Ortelius (nach 1570)
 Franciæ orientalis (vulgo Franckenlant) descriptio / Auctore Sebast. a Ro-
 tenhan. Antwerpen: Abraham Ortelius, zwischen 1570 und 1612. - Kupferstich,
 koloriert, 24,8 x 36,2 cm. - Maßstab ca. 1:590.000.
 Signatur: Universitätsbibliothek Würzburg 36/A 20.39

bis zum Jahr 1612 zahlreiche Auflagen.

Anhang A. Karten

Die Frankenkarte Rotenhans ist im Hochformat eingefügt. Dadurch ist die eigentlich nach Süden ausgerichtete Karte geostet, d.h. der 'Oriens' liegt oben. Die Karte wird dominiert vom Lauf des Mains, der im Fichtelgebirge entspringt und bis Frankfurt am unteren Kartenrand fließt. Links unten ist ein Meilenmessstab angebracht. Das vorliegende Stück ist handkoloriert.

A.2.3. Franckenlandt = Francia orientalis / Gerhard Mercator (1600)



Abb. A.10.: Gerhard Mercator (1600)
Franckenlandt = Francia orientalis / Per Gerardum Mercatorem – o.O., ca. 1600. - Kupferstich, koloriert, 47,5 x 31,8 cm. - Maßstab ca. 1:700.000.
Signatur: Universitätsbibliothek Würzburg 36/A 10.25

Eine Karte des Fränkischen Reichskreises, beschrieben auf Franconica-Online (Franconica-Online Historische Karten, 2018) als:

Gerhard Mercator (eigentlich Kremer, 1512-94) gilt als der bedeutendste Kartograph nach Ptolemäus. Im Gegensatz zu vielen seiner Zeitgenossen schuf er fast alle seine Karten selbst. Er stammt aus Rupelmonde in Flandern, siedelte aber 1552 nach Duisburg um, wo er seitdem arbeitete. Er wurde Kosmograph des Herzogs Wilhelm von Jülich-Cleve-Berg.

Von 1585 an begann sein berühmter 'Atlas' zu erscheinen, der diesen Namen für ein Kartenwerk populär machte. Der Deutschlandteil (*Germaniae tabulae geographicae*) enthält eine Frankenkarte, die im Gegensatz zum Ortelius-Atlas von Mercator selbst neu entworfen wurde. Diese Karte wurde in zahlreichen Auflagen auch nach Mercators Tod verbreitet. Die Karte ist genordet ('Septentrio' am oberen Rand) und zeigt daher das uns gewohnte Bild. Ihre Ausdehnung reicht von Schlitz-Vacha-Schmalkalden-Saalfelden bis Ellwangen-Pappenheim-Kelheim-Straubing bzw. in West-Ost-Richtung von Hanau bis Cham. Der Lauf des Mains einschließlich seiner Nebenflüsse sowie die Schreibweise der Ortsnamen lassen darauf schließen, dass Rotenhans Karte als Vorlage gedient hat.

A.2.4. *Nova Franconiae descriptio* / Abraham Goos (1626)

Eine Karte des Fränkischen Reichskreises, beschrieben auf [Franconica-Online](#) ([Franconica-Online Historische Karten](#), 2018) als:

Abraham Goos (um 1590-1643) arbeitete seit 1614 als Graveur in Amsterdam. Im 'Atlantis Maioris Appendix, Sive Pars Altera', den Jan Jansson d.Ä. 1630 herausgab, befindet sich die von ihm gestochene Frankenkarte. Sie umfasst in der uns geläufigen nördlichen Ausrichtung das Gebiet vom Thüringer Wald bis Hohenlohe, bzw. vom Frankfurter Raum bis Hof.

Eingerahmt wird das Kartenbild von Ansichten der Städte Nürnberg, Würzburg, Fulda, Rothenburg o.d.T. und Büdingen sowie einer Planansicht von Bamberg. Außerdem illustrieren vier Figuren die fränkische

Anhang A. Karten

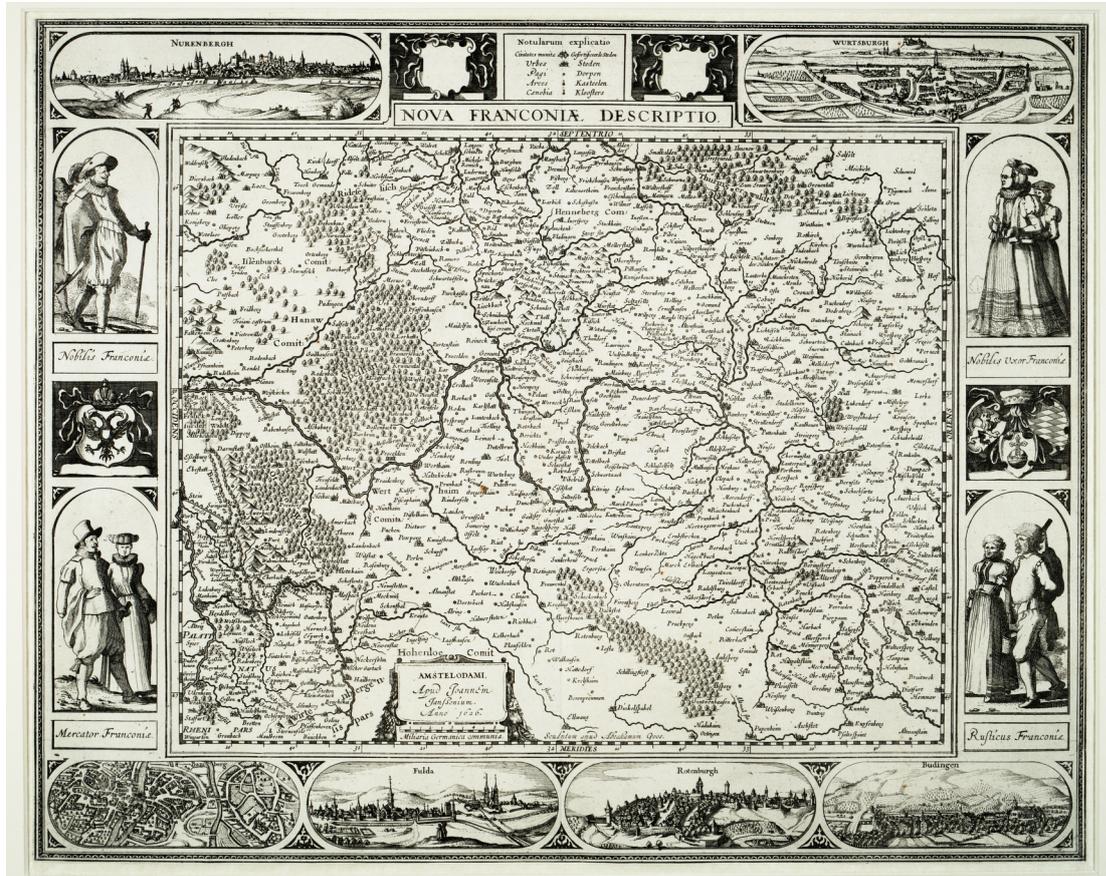


Abb. A.11.: Abraham Goos (1626)
Nova Franconiae descriptio / Sculptum apud Abrahamum Goos. - Amsterdam: Joannes Janſonius, 1626. - Kupferstich, 40 x 32 cm, Bildgröße 54,4 x 43,5 cm. - Maßstab 1: 350.000.
Signatur: Universitätsbibliothek Würzburg 36/G.f.m.9-12,139

Tracht, nämlich ein adliges Ehepaar (Nobilis), ein Kaufmann mit Frau (Mercator) und ein Bauernpaar (Rusticus).

A.2.5. Franconia : Neue Carten Des Gantzen Fränckischen Krayses / Matthäus Merian (1656)

Eine Karte des Fränkischen Reichskreises, beschrieben auf Franconica-Online (Franconica-Online Historische Karten, 2018) als:

A.2. Fränkischer Reichskreis



Abb. A.12.: Matthäus Merian (1656).

Franconia : Neue Carten Des Gantzen Fränckischen Krayses (aus: Topographia Franconiae... , 2. Auflage) / [M. Merian]. - Frankfurt am Main, 1656. - Kupferstich, 33,5 x 25,8 cm. - Maßstab: ca. 1:720.000
 Signatur: Universitätsbibliothek Würzburg 36/B 13.150

Matthäus Merian (1593-1650) stammt aus Basel. Nach ausgedehnten Lehr- und Wanderjahren in Frankreich und den Niederlanden ließ er sich endlich etwa 1625 in Frankfurt am Main nieder. Dort gründete er ein Verlagshaus, das nach seinem Tod von seinem Sohn Matthäus d.J. (1621 -87) fortgeführt wurde.

Bekannt ist Merian vor allem wegen seiner Topographien, einer Sammlung deutscher Orts- und Landesbeschreibungen von insgesamt 30 Bänden mit über 2000 Städteansichten. Die Texte dazu verfasste der Ulmer Hofmeister Martin Zeiller (1589-1661). In Merians Verlag er-

schienen auch andere Werke, darunter das 'Theatrum Europaeum', das zeitgeschichtlich orientiert war. Im 2. Band, der 1633 erschien (Textverfasser ebenfalls Zeiller) ist die erste Frankenkarte Merians enthalten. Nachstiche mit kleineren Veränderungen finden sich in den Ausgaben der Topographia Franconiae (1. Ausgabe 1648, 2. Ausgabe 1656). Aus der zweiten Ausgabe stammt der hier gezeigte Stich. Beim inhaltlichen Vergleich mit der Frankenkarte Blaeus ergeben sich weitgehende Übereinstimmungen.

A.2.6. Circulus Franconicus / Matthias Seutter (1731)

Eine Karte des Fränkischen Reichskreises, beschrieben auf Franconica-Online (Franconica-Online Historische Karten, 2018) als:

Matthias (auch: Matthäus) Seutter (1678-1757), ein Schüler Homanns, eröffnete 1707 in Augsburg eine kartographische Firma. Nach seinem Tod führten sein Sohn Albrecht und seine Schwiegersöhne Lotter und Probst das Geschäft weiter. Seine Karte des fränkischen Reichskreises erschien 1731. Sie bildet Blatt 69 in seinem großen Atlas.

Der erste Eindruck ist der einer der üblichen flächenkolorierten Karten. Die Besonderheit bei Seutters Karte ist jedoch das Gitternetz aus Längen- und Breitengraden, das das gesamte Kartenbild überzieht. In der Randleiste sind Orientierungshilfen in Form von Groß- und Kleinbuchstaben angebracht. Mit Hilfe eines zur Karte gehörenden Registers ist es möglich, die ausgewiesenen geographischen Bezeichnungen gezielt aufzusuchen. Die Orte, differenziert nach ihrer Größe, sind durch Hauptstraßen (*viae publicae*) verbunden, die die Karte ebenfalls ausweist. Kleine Posthörner bezeichnen die Post-Relaisstationen (z.B. bei Ansbach).

Damit erweist sich Seutter als Kartograph, der für ein reiseorientiertes Publikum arbeitete. Seine Übersichtskarte ist für solche Zwecke wohl die beste in der ersten Hälfte des 18. Jahrhunderts. Trotzdem dachte

A.2. Fränkischer Reichskreis

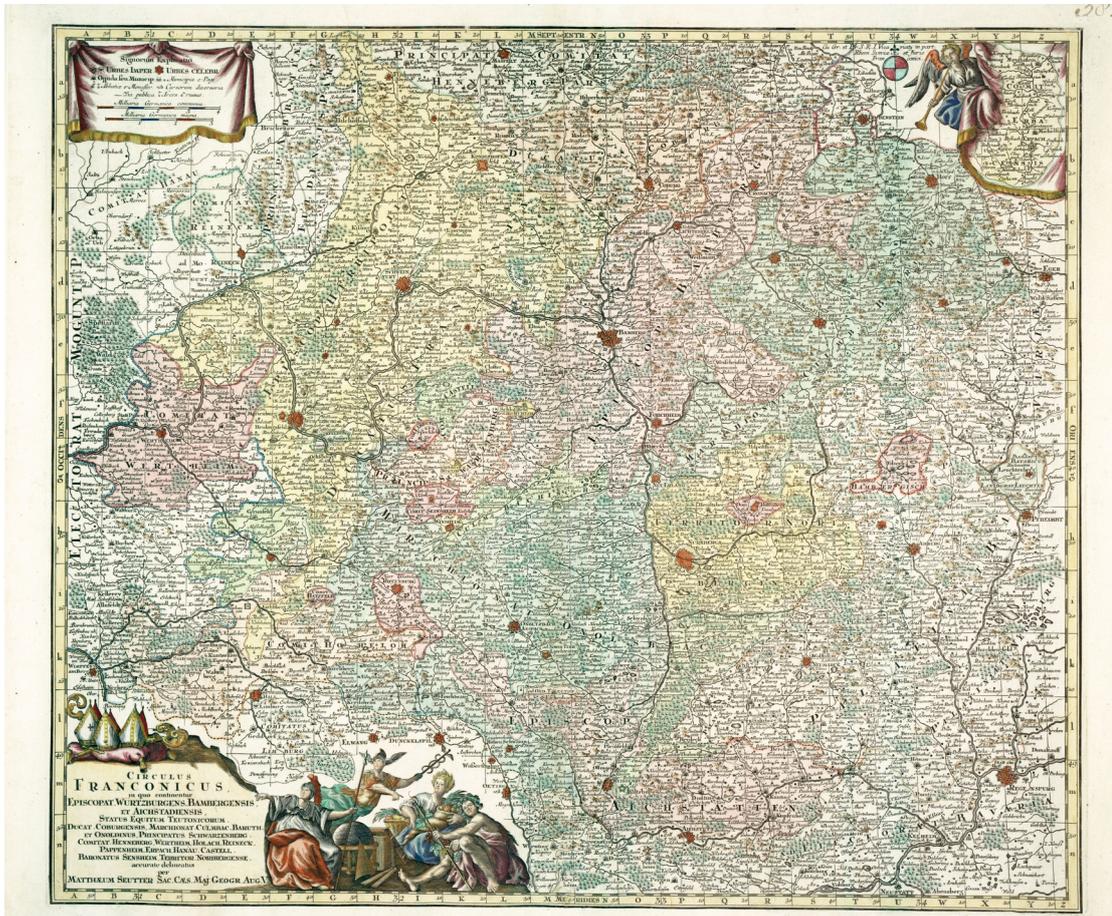


Abb. A.13.: Beschreibung Matthias Seutter (1731)

Circulus Franconicus : in quo continentur Episcopatus Würtzburgensis, Bambergensis, et Aichstadiensis. Status Equitum Teutonicorum, Ducatus Coburgensis, Marchionatus Culmbac. Baruth. Et Onoldinus, Principatus Schwarzenberg, Comitatus Henneberg, Wertheim, Holach, Reineck, Pappenheim, Erpach, Hanau, Castell, Baronatus Sensheim, Territorium Noribergense / accurate delineatus per Matthaeum Seutter. – Augsburg, (1731). - Kupferstich, koloriert, 55,3 x 47,4 cm. - Maßstab ca. 1: 450 000. Signatur: Universitätsbibliothek Würzburg 36/A 1.18

er auch an repräsentative Ausgestaltung: In der Titelkartusche wird Franken allegorisch dargestellt durch zwei Figuren mit Feldfrüchten und Weinreben.

A.2.7. Das Fürstenthum Würzburg / Fackenhofen (1791/1805)

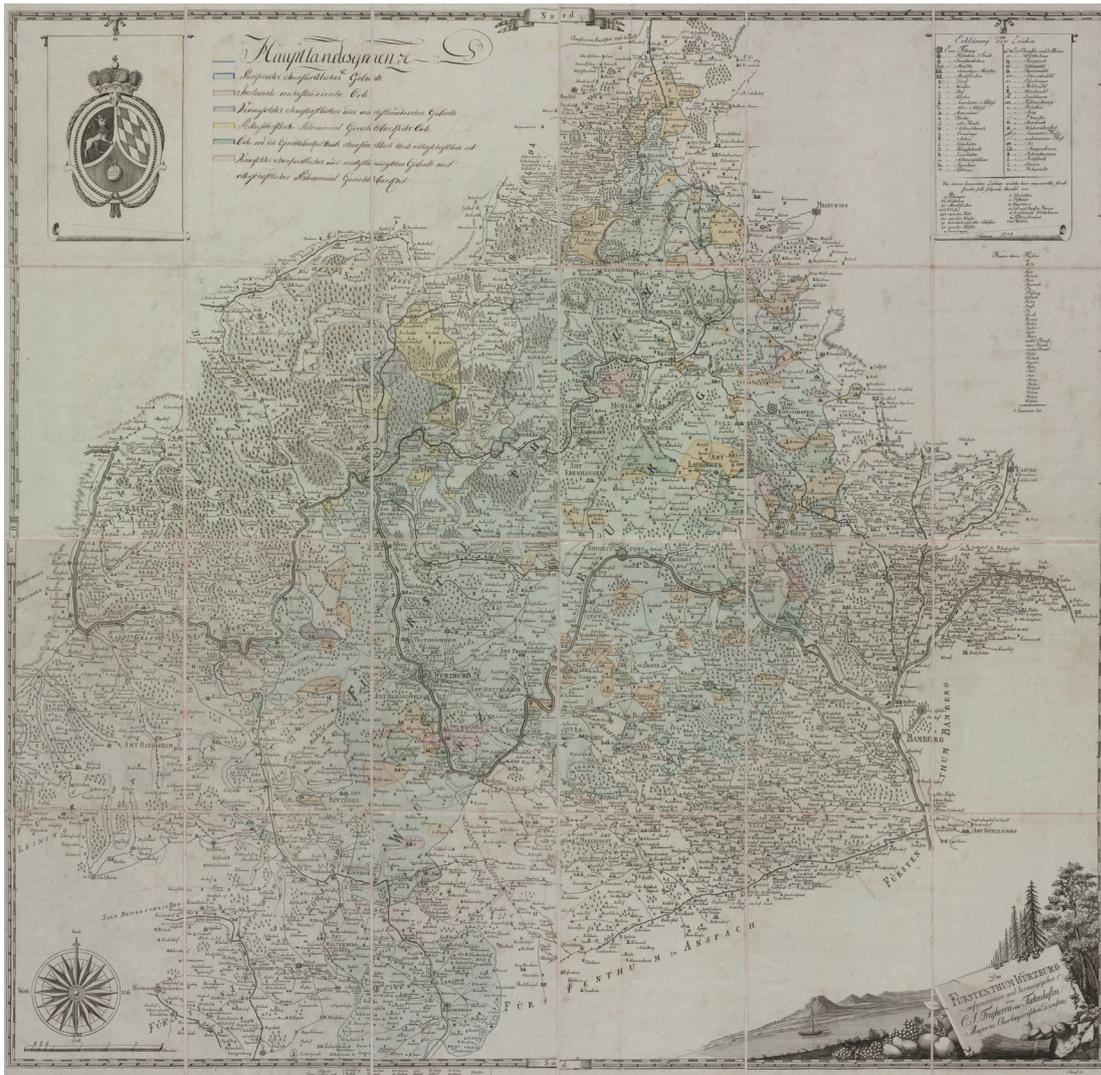


Abb. A.14.: Fackenhofen (1791/1805)

Das Fürstenthum Würzburg. Aufgenommen und herausgegeben von C. J. Freyherrn von Fackenhofen, Major in Churbayerischen Diensten, Stumpf sc. Würzburg 1805. – Kupferstich, koloriert, 100 x 99 cm, Faltgröße 17 x 25 cm. Signatur: Universitätsbibliothek Würzburg 36/A 50.8

Eine Karte des Fränkischen Reichskreises, beschrieben auf Franconica-Online (Franconica-Online Historische Karten, 2018) als:

Auf Betreiben des Würzburgischen Obristen und Hofkriegsrats Christoph Friedrich Seelmann wurde 1786 der Oberleutnant Carl von Fackenhofen (1745-1804) mit der Neuerarbeitung einer Karte des Hochstifts Würzburg beauftragt, die insbesondere für die Planung von Truppenbewegungen und Einquartierungen alle wichtigen Informationen korrekt wiedergeben sollte. Die bislang dafür gebrauchten Karten aus dem Verlag von Johann Baptist Homann hatten sich in der Praxis als sehr fehlerhaft erwiesen. Erst 1805, nach der Säkularisation des Hochstifts unter kurpfälzbayerischem Regime, wurde eine Druckausgabe, gestochen von dem Würzburger Kupferstecher Christoph Joseph Stumpf (1754-1809), hergestellt.

Die Karte zeigt „die einzig authentische, weil offizielle Darstellung des Würzburger Staatsgebiets an der Schwelle des 19. Jahrhunderts. Darin liegt bei der Unsicherheit der Herrschafts- und Hoheitsbegriffe der Zeit und den zahllosen Streitigkeiten um die sich vielfach und vielschichtig überschneidenden Rechtsbereiche ihr einzigartiger Wert (H. H. Hofmann). Die Karte weist links oben das kurpfälzbayerische Landeswappen auf, links unten Kompassrose und Stundenmaßstab, rechts unten als Staffage eine Landschaftsszene mit dem Main und dem Kartentitel. Die Legende rechts oben bietet eine Erklärung der 48 verwendeten kartographischen Symbole, von 'Festung' und 'Residenzstadt' über 'Posthaus', 'Chaussée' und 'Wirtshaus' bis 'Bildstock', 'Kreuz' und 'Hohgericht'. Die 'Summa' der solchermaßen auf der Karte angegebenen Orte beträgt allerdings nicht 3709, wie die Legende verzeichnet, sondern 3739. Die veröffentlichte Karte weist gegenüber dem heute im Staatsarchiv Würzburg (Würzburger Risse und Pläne I/1) aufbewahrten gezeichneten Original manche Vereinfachungen und Auslassungen auf. Der Stecher Stumpf hat offenkundig auch um eines ästhetisch stimmigeren Schriftbildes willen einige Ortspunkte leicht versetzt. Dennoch ist sie ein stimmiges Abbild der politischen Verhältnisse in Mainfranken am Ende des Alten Reiches, allerdings durch die politische Brille des Würzburger Hochstifts gesehen. Literatur: Hans Hubert Hofmann: Die Würzburger Hochstiftskarte des Oberleutnants von Fackenhofen (1791), Würzburg 1956 (Zitat oben S. 18) Herbert Schott, Die Entstehung der sog. Fackenhofen-Karte des Hochstifts Würzburg. In: Mainfränkisches

Jahrbuch für Geschichte und Kunst 2006, S. 129-148.

A.2.8. Franconia vulgo Franckenlandt / Willem Janszoon Blaeu (1650)



Abb. A.15.: Willem Janszoon Blaeu (1650)
Franconia vulgo Franckenlandt / Excudit Guiljelmus Blaeu. – Amsterdam,
ca. 1650. - Kupferstich, koloriert, 47,3 x 36 cm. – Maßstab ca. 1:600.000.
Signatur: Universitätsbibliothek Würzburg 36/A 10.19

Eine Karte des Fränkischen Reichskreises, beschrieben auf Franconica-Online
(Franconica-Online Historische Karten, 2018) als:

Willem Janszoon (1571-1638) aus Amsterdam, der sich seit 1621 zur Unterscheidung von seinem Konkurrenten Jan Jansson 'Blaeu' nannte, war ein Schüler des berühmten dänischen Astronomen Tycho Brahe.

1633 wurde er 'Kartenmaker' der Ostindischen Kompanie und betrieb in der Folgezeit zusammen mit seinen Söhnen Joan und Cornelis bald die führende kartographische Firma in den Niederlanden. Von verschiedenen Stechern übernahm er kartographische Materialien, stach sie meist um und brachte sie seit 1635 in Atlanten heraus.

Die Karte stammt aus einer holländischen Ausgabe seines 'Novus Atlas', 'Toonnael des aerdriicx', und ist auf der Rückseite mit Erläuterungstext bedruckt. Das Blatt ist handkoloriert, die Grenzen Frankens sind von Hand farbig eingetragen. Es handelt sich offenbar um einen leicht vergrößerten Kartengrundriss von Mercator mit neu hinzugefügten Signaturen und Ortsnamen. An die Stelle der Titeltartusche Mercators setzte Blaeu einen senkrechten Meilenstab und nutzte den dadurch freiwerdenden Platz links unten für seinen neuen Titel. Beim Vergleich mit der Frankenkarte Merians ergeben sich weitgehende Übereinstimmungen.

A.2.9. Carte Topographique D'Allemagne / Daniel Adam Hauer (1787)

Eine Karte des Fränkischen Reichskreises, beschrieben auf Franconica-Online (Franconica-Online Historische Karten, 2018) als:

Blatt 41 aus der Kartenserie eine 'Carte topographique d'Allemagne' des Frankfurter Verlages Jaeger schließt sich nördlich an Blatt 50 an. Das Kartenbild umfasst das Gebiet zwischen Kloster Ebrach im Süden und Salzungen im Norden, Fulda im Westen und Kulmbach im Osten. Die einzelnen Territorien sind im Titel aufgeführt.

Der Stecher Daniel Adam Hauer (geb. 1734) aus Nürnberg hat seine Karte nach einer Vorlage der nicht ermittelten Zeichner „I. & K.“ gestochen. Maßstab und Gradeinteilung stimmen mit Blatt 50 völlig überein. Im Unterschied dazu sind jedoch die geographischen und politischen Bezeichnungen durchwegs deutsch.

Anhang A. Karten

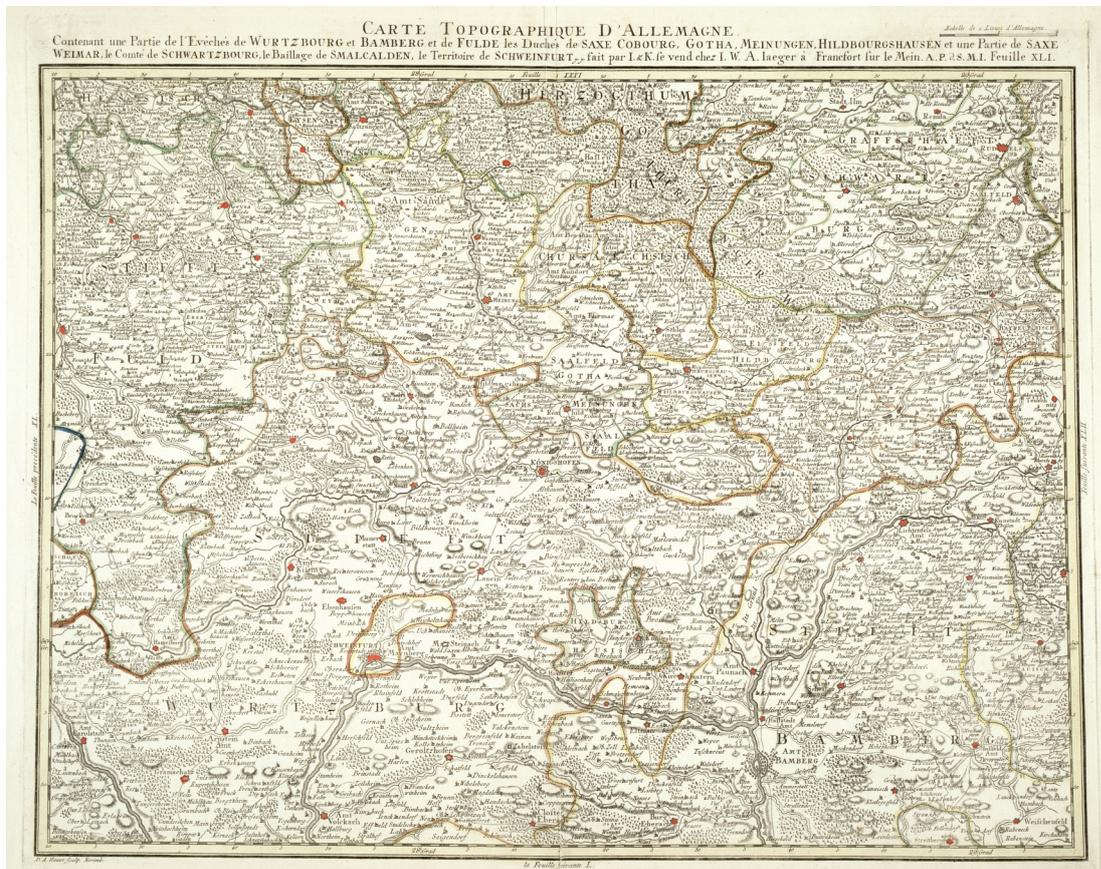


Abb. A.16.: Daniel Adam Hauer (1787)
Carte Topographique D'Allemagne Contenant une Partie de l'Evêchés de Wurtzbourg et Bamberg et Fulde, les Duchés de Saxe Cobourg, Gotha, Meiningen, Hildbourgshausen et une Partie de Saxe Weimar, le Comté de Schwartzbourg, le Baillage de Smalcalden, le Territoire de Schweinfurt : A. P. d. S. M. I. / fait par I. & K. D. A. Hauer sculp. Norimb.
Francfort sur le Mein: I. W. A. Jaeger, (ca. 1787). - Kupferstich, koloriert, 60,5 x 45,2 cm
Signatur: Universitätsbibliothek Würzburg 36/A 1.16-41.

A.2.10. Circulus Franconicus / Matthaeus Seutter (1731)

Eine Karte des Fränkischen Reichskreises, beschrieben auf David Rumsey Map Collection (Rumsey Historical Map Collection, 2018) als:

This is the first issue of Seutter's Atlas Novus with the first issue of the title page and first issues of many of the maps. Page numbers of maps

A.2. Fränkischer Reichskreis

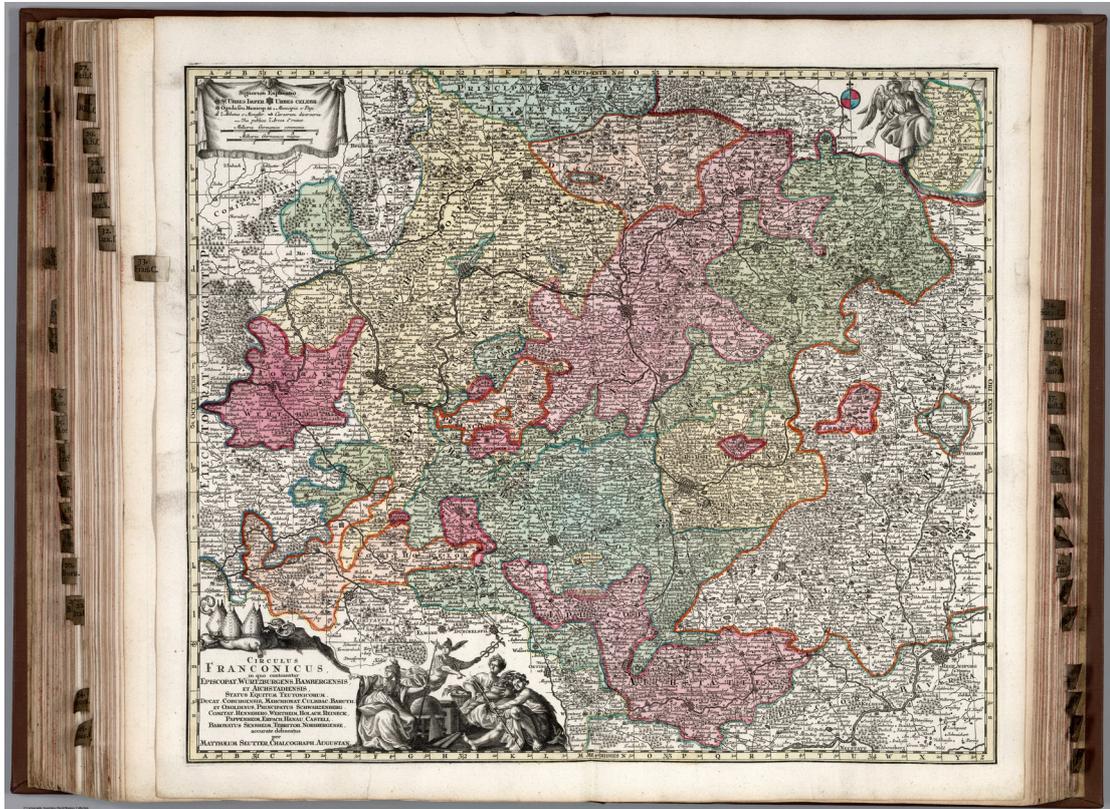


Abb. A.17.: Matthaeus Seutter (1731)

Atlas Novus sive Tabulae Geographicae Totius Orbis Faciem, Partes, Imperia, Regna et Provincias Exhibentes Exactissima Cura Juxta Recentissimas Observation Aeri Incisae et Venum Expositae a Matthaeo Seutter Chalcogr. Augustae Vindelicorum.

Signatur: David Rumsey Historical Map Collection 12041.110.

shown on tabs - missing ones shown in parentheses. Corresponding alphabetical index sheets numbered in Full Title field to match map numbers. Color. Relief shown by sketches. In German and/or Latin.

Literaturverzeichnis

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*.

ABBYY FineReader 14 (2018). OCR-Software mit PDF-Bearbeitung | ABBYY FineReader 14. <https://www.abbyy.com/de-de/finereader/>. (zuletzt abgerufen am 17.06.2018).

Afzal, M. Z., Krämer, M., Bukhari, S. S., Yousefi, M. R., Shafait, F., und Breuel, T. M. (2013). Robust binarization of stereo and monocular document images using percentile filter. In *International Workshop on Camera-Based Document Analysis and Recognition*, pages 139–149. Springer.

Ahmed, S., Weber, M., Eichenberger-Liwicki, M., und Dengel, A. (2011). Text/Graphics Segmentation in Architectural Floor Plans. In *International Conference on Document Analysis and Recognition*. IEEE Computer Society Conference Publications Operations Committee.

Arganda-Carreras, I., Sorzano, C. O., Marabini, R., Carazo, J. M., Ortiz-de Solorzano, C., und Kybic, J. (2006). Consistent and elastic registration of histological sections using vector-spline regularization. In *International Workshop on Computer Vision Approaches to Medical Image Analysis*, pages 85–95. Springer.

argman (2018). A tensorflow implementation of EAST text detector. <https://github.com/argman/EAST>. (zuletzt abgerufen am 24.06.2018).

Literaturverzeichnis

- Bay, H., Ess, A., Tuytelaars, T., und Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359.
- Bergstra, J. und Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.
- Bhowmik, S., Sarkar, R., Nasipuri, M., und Doermann, D. (2018). Text and non-text separation in offline document images: a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, 21(1):1–20.
- Biasi, F. (2013). Google lat long: National geographic shares rich map content with the world via google maps engine. http://google-latlong.blogspot.ca/2013/12/national-geographic-shares-rich-map_6.html. abgefufen am 02.11.2014.
- Breuel, T. M., Ul-Hasan, A., Al-Azawi, M. A., und Shafait, F. (2013). High-Performance OCR for Printed English and Fraktur Using LSTM Networks. In *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, ICDAR '13*, pages 683–687. IEEE Computer Society.
- Briechele, K. und Hanebeck, U. D. (2001). Template matching using fast normalized cross correlation. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 95–102. International Society for Optics and Photonics.
- Burger, W. und Burge, M. J. (2009). *Principles of Digital Image Processing: Fundamental Techniques*. Springer Publishing Company, Incorporated, 1 edition.
- Cao, R. und Tan, C. L. (2002). Text/Graphics Separation in Maps. In *Selected Papers from the Fourth International Workshop on Graphics Recognition Algorithms and Applications, GREC '01*, pages 167–177, London, UK, UK. Springer-Verlag.
- Chen, C.-C., Knoblock, C. A., Shahabi, C., Chiang, Y.-Y., und Thakkar, S. (2004). Automatically and accurately conflating orthoimagery and street maps. In *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pages 47–56. ACM.
- Chen, H., Tsai, S. S., Schroth, G., Chen, D. M., Grzeszczuk, R., und Girod, B.

- (2011). Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2609–2612. IEEE.
- Chen, X. und Yuille, A. (2004). Detecting and reading text in natural scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.
- Chiang, Y.-Y. und Knoblock, C. A. (2010a). An Approach for Recognizing Text Labels in Raster Maps. In *Proceedings of the 20th International Conference on Pattern Recognition*, pages 3199–3202.
- Chiang, Y.-Y. und Knoblock, C. A. (2010b). Strabo: A System for Extracting Road Vector Data from Raster Maps. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10*, pages 544–545, New York, NY, USA. ACM.
- Chiang, Y.-Y. und Knoblock, C. A. (2015). Recognizing text in raster maps. *GeoInformatica*, 19(1):1–27.
- Chiang, Y.-Y., Leyk, S., und Knoblock, C. A. (2011). Integrating color image segmentation and user labeling for efficient and robust graphics recognition from historical maps. In *Proceedings of the Ninth IAPR International Workshop on Graphics Recognition*.
- Chiang, Y.-Y., Leyk, S., und Knoblock, C. A. (2014). A survey of digital map processing techniques. *ACM Computing Surveys (CSUR)*, 47(1):1.
- Craswell, N. (2009). *Mean Reciprocal Rank*, pages 1703–1703. Springer US, Boston, MA.
- Deng, D., Liu, H., Li, X., und Cai, D. (2018). PixelLink: Detecting Scene Text via Instance Segmentation. *arXiv preprint arXiv:1801.01315*.
- Dhar, D. und Chanda, B. (2006). Extraction and recognition of geographical features from paper maps. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4):232–245.

Literaturverzeichnis

- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.
- Dori, D. und Wenyin, L. (1999). Automated CAD conversion with the machine drawing understanding system: concepts, algorithms, and performance. *IEEE Transactions on Systems, Man, and Cybernetics-part A: systems and humans*, 29(4):411–416.
- Doucette, P., Kovalerchuk, B., Brigantic, R., Seedahmed, G., und Graff, B. (2007). A methodology for automated vector-to-image registration. In *Applied Imagery Pattern Recognition Workshop, 2007. AIPR 2007. 36th IEEE*, pages 9–14. IEEE.
- Fleet, C., Kowal, K., und Pridal, P. (2012). Georeferencer: Crowdsourced Georeferencing for Map Library Collections. *D-Lib Magazine*, 18(11):5.
- Fletcher, L. und Kasturi, R. (1988). A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:910–918.
- Franconica-Online Historische Karten (2018). Franconica-Online Historische Karten. <http://franconica.uni-wuerzburg.de/ub/topographia-franconiae/maps.html>. (zuletzt abgerufen am 13.07.2018).
- Fritzen, P. und Solchenbach, K. (2017). Die Ortelius-Karte von Luxemburg und ihre Zustände. *Hémecht: Zeitschrift für Luxemburger Geschichte*.
- Gatos, B., Pratikakis, I., und Perantonis, S. J. (2005). Text detection in indoor/outdoor scene images. In *First International Workshop on Camera-based Document Analysis and Recognition*.
- Girshick, R. B., Donahue, J., Darrell, T., und Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524.
- Gllavata, J., Ewerth, R., und Freisleben, B. (2004). Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In *Pattern*

- Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 425–428. IEEE.
- Glorot, X., Bordes, A., und Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.
- Google Fonts (2018). Google Fonts. <https://fonts.google.com/>. (zuletzt abgerufen am 2.06.2018).
- Google Maps Engine (2016). Google Maps Engine - Google. <https://mapsengine.google.com/about/index.html>. (zuletzt abgerufen am 04.12.2017).
- Graves, A., Fernández, S., Gomez, F., und Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM.
- Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., und Schmidhuber, J. (2009). A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868.
- He, K., Zhang, X., Ren, S., und Sun, J. (2015). Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385.
- He, W., Zhang, X.-Y., Yin, F., und Liu, C.-L. (2017). Deep direct regression for multi-oriented scene text detection. *arXiv preprint arXiv:1703.08289*.
- Hochreiter, J. (1991). Untersuchungen zu dynamischen neuronalen Netzen. *Diploma, Technische Universität München*, 91:1.
- Hochreiter, S. und Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Höhn, W. (2013). Detecting Arbitrarily Oriented Text Labels in Early Maps. In Sanches, J. a. M., Micó, L., und Cardoso, J. S., editors, *Pattern Recognition*

- and Image Analysis*, volume 7887 of *Lecture Notes in Computer Science*, pages 424–432. Springer Berlin Heidelberg.
- Höhn, W. (2017). Deep Learning for Place Name OCR in Early Maps. In *Proceedings of the 2nd International Workshop on Exploring Old Maps*.
- Höhn, W., Schmidt, H.-G., und Schöneberg, H. (2013). Semiautomatic recognition and georeferencing of places in early maps. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries, JCDL '13*, pages 335–338, New York, NY, USA. ACM.
- Höhn, W. und Schommer, C. (2016a). Annotating and Georeferencing of Digitized Early Maps. In *Digital Humanities 2016: Conference Abstracts*.
- Höhn, W. und Schommer, C. (2016b). RAT: A Referencing and Annotation Tool for Digitized Early Maps. http://www.dhbenelux.org/wp-content/uploads/2016/05/70_Hoehn-Schommer_FinalAbstract_DHBenelux_short.pdf. DH Benelux Conference 2016.
- Höhn, W. und Schommer, C. (2017a). Georeferencing of Place Markers in Digitized Early Maps by Using Similar Maps as Data Source. In *Digital Humanities 2017: Conference Abstracts*.
- Höhn, W. und Schommer, C. (2017b). RAT 2.0. In *Digital Humanities 2017: Conference Abstracts*.
- Howard, B. C. (2014). National Geographic Thematic Maps Launch in New Google Tool. <http://news.nationalgeographic.com/news/2014/02/140227-google-maps-gallery-maps-engine/>. (zuletzt abgerufen am 06.11.2017).
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE CVPR*.
- Jenny, B. und Hurni, L. (2011). Cultural heritage: Studying cartographic heritage: Analysis and visualization of geometric distortions. *Computers & Graphics*, 35:402–411.

- Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V. R., Lu, S., et al. (2015). ICDAR 2015 competition on robust reading. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1156–1160. IEEE.
- Kasturi, R., Bow, S. T., El-Masri, W., Shah, J., Gattiker, J. R., und Mokate, U. B. (1990). A system for interpretation of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:978–992.
- Kim, K., Cheon, Y., Hong, S., Roh, B., und Park, M. (2016). PVANET: deep but lightweight neural networks for real-time object detection. *CoRR*, abs/1608.08021.
- Kim, K., Jung, K., und Kim, J. (2003). Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1631–1639.
- Klöti, T. (1994). *Johann Friedrich von Ryhiner, 1732-1803: Berner Staatsmann, Geograph, Kartenbibliograph und Verkehrspolitiker*. Geographische Gesellschaft, Bern.
- Kovalerchuk, B. und Chase, R. (2005). Image analysis method for overcoming source distortion using algebraic invariant methods. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI*, volume 5806, pages 371–382. International Society for Optics and Photonics.
- Krizhevsky, A., Sutskever, I., und Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- LeCun, Y., Bengio, Y., und Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.
- Lee, D.-T. und Schachter, B. J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242.

Literaturverzeichnis

- Lewis, J. (1995). Fast template matching. In *Vision Interface*, volume 95, pages 15–19.
- Liao, M., Shi, B., und Bai, X. (2018). TextBoxes++: A Single-Shot Oriented Scene Text Detector. *arXiv:1801.02765 [cs]*. arXiv: 1801.02765.
- Liao, M., Shi, B., Bai, X., Wang, X., und Liu, W. (2016). TextBoxes: A Fast Text Detector with a Single Deep Neural Network. *arXiv:1611.06779v1 [cs]*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., und Berg, A. C. (2015). SSD: single shot multibox detector. *CoRR*, abs/1512.02325.
- Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., und Yan, J. (2018). FOTS: Fast Oriented Text Spotting with a Unified Network. *arXiv:1801.01671 [cs]*. arXiv: 1801.01671.
- Liu, Y. und Jin, L. (2017). Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection. *CoRR*, abs/1703.01425.
- Lladós, J., Valveny, E., Sánchez, G., und Martí, E. (2002). Symbol Recognition: Current Advances and Perspectives. In Blostein, D. und Kwon, Y.-B., editors, *Graphics Recognition Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 104–128. Springer Berlin Heidelberg.
- Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99*, pages 1150–, Washington, DC, USA. IEEE Computer Society.
- Moyssset, B., Kermorvant, C., und Wolf, C. (2017). Full-Page Text Recognition: Learning Where to Start and When to Stop. *arXiv preprint arXiv:1704.08628*.
- NGA About (2018). About NGA. <https://www.nga.mil/About/Pages/Default.aspx>. (zuletzt abgerufen am 18.07.2018).
- Ocropy Github (2018). tmbdev/ocropy: Python-based tools for document ana-

- lysis and OCR. <https://github.com/tmbdev/ocropy>. (zuletzt abgerufen am 18.06.2018).
- OldMapsOnline (2018). Collections - Old Maps Online Project. <http://www.oldmapsonline.org/about/>. (zuletzt abgerufen am 24.01.2018).
- OmniPage OCR (2018). OmniPage OCR- und Scansoftware Produktübersicht. https://shop.nuance.de/store/nuanceeu/de_DE/Content/pbPage.microsite-omnipage-overview. (zuletzt abgerufen am 17.06.2018).
- OpenCV Template Matching (2018). Template Matching - OpenCV 2.4.13.6 documentation. https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html. (zuletzt abgerufen am 28.06.2018).
- Postel, H. J. (1969). Die Kölner Phonetik—Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse. *IBM-Nachrichten*, 19:925–931.
- QGIS Georeferenzier Plugin (2018). Georeferenzier Plugin. https://docs.qgis.org/2.18/de/docs/user_manual/plugins/plugins_georeferencer.html.
- Rath, T. M. und Manmatha, R. (2007). Word Spotting for Historical Documents. *International Journal of Document Analysis and Recognition (IJ DAR)*, 9(2-4):139–152.
- Redmon, J., Divvala, S., Girshick, R., und Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Ronneberger, O., Fischer, P., und Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- Roy, P. P., Lladós, J., und Pal, U. (2007). Text/Graphics Separation in Color Maps. In *Computing: Theory and Applications, 2007. ICCTA'07. International Conference on*, pages 545–551. IEEE.

Literaturverzeichnis

- Roy, P. P., Vazquez, E., Lladós, J., Baldrich, R., und Pal, U. (2008). A System to Segment Text and Symbols from Color Maps. In Liu, W., Lladós, J., und Ogier, J.-M., editors, *Graphics Recognition. Recent Advances and New Opportunities*, pages 245–256. Springer-Verlag, Berlin, Heidelberg.
- Rumsey Historical Map Collection (2018). David Rumsey Historical Map Collection | Collection History. <https://www.davidrumsey.com/about>. (zuletzt abgerufen am 04.06.2018).
- Rusiñol, M., Aldavert, D., Toledo, R., und Lladós, J. (2015). Efficient Segmentation-free Keyword Spotting in Historical Document Collections. *Pattern Recognition*, 48(2):545–555.
- Sabir, E., Rawls, S., und Natarajan, P. (2017). Implicit Language Model in LSTM for OCR. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 7, pages 27–31. IEEE.
- Schalansky, J. (2008). *Fraktur Mon Amour*. Princeton Architectural Press.
- Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676.
- Schmidt, H.-G. (2007). Franconica-Online. <http://franconica.uni-wuerzburg.de/ub/topographia-franconiae/uebertopographia.html>. (zuletzt abgerufen am 06.11.2017).
- Seedahmed, G. und Martucci, L. (2002). Automated image registration using geometrical invariant parameter space clustering (GIPSC). *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):318–323.
- Shi, B., Bai, X., und Yao, C. (2016). An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99).
- Shi, Z., Setlur, S., und Govindaraju, V. (2005). Text extraction from gray scale

- historical document images using adaptive local connectivity map. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition, ICDAR '05*, pages 794–798, Washington, DC, USA. IEEE Computer Society.
- Shivakumara, P., Phan, T., und Tan, C. (2009). A robust wavelet transform based technique for video text detection. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 1285–1289. IEEE.
- Simard, P. Y., Steinkraus, D., und Platt, J. C. (2003). Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2, ICDAR '03*, pages 958–962. IEEE Computer Society.
- Simon, R., Barker, E., Isaksen, L., und de Soto Cañamares, P. (2015). Linking Early Geospatial Documents, One Place at a Time: Annotation of Geographic Documents with Recogito. *e-Perimtron*, 10(2):49–59.
- Simon, R., Barker, E., Isaksen, L., und de Soto Cañamares, P. (2017). Linked Data Annotation Without the Pointy Brackets: Introducing Recogito 2. *Journal of Map & Geography Libraries*, 13(1):111–132.
- Simon, R., Haslhofer, B., Robitza, W., und Momeni, E. (2011). Semantically augmented annotations in digitized map collections. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries, JCDL '11*, pages 199–202, New York, NY, USA. ACM.
- Simonyan, K. und Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Smith, R. (2007). An overview of the Tesseract OCR engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 629–633. IEEE.
- Smith, R. (2016). Tesseract blends old and new ocr technology. https://github.com/tesseract-ocr/docs/tree/master/das_tutorial2016. (zuletzt abgerufen am 17.06.2018).

Literaturverzeichnis

- Smith, R., Gu, C., Lee, D.-S., Hu, H., Unnikrishnan, R., Ibarz, J., Arnaud, S., und Lin, S. (2016). End-to-end interpretation of the french street name signs dataset. In *European Conference on Computer Vision*, pages 411–426. Springer.
- Spektrum der Wissenschaft (2000a). geodätisches Datum. <http://www.spektrum.de/lexikon/geowissenschaften/geodaetisches-datum/5600>. (zuletzt abgerufen am 06.11.2017).
- Spektrum der Wissenschaft (2000b). Ortsring - Lexikon der Geowissenschaften. <http://www.spektrum.de/lexikon/geowissenschaften/ortsring/11728>. (zuletzt abgerufen am 28.07.2017).
- Springmann, U. (2015). Ocrocis: A high accuracy OCR method to convert early printings into digital text—A Tutorial. <http://cistern.cis.lmu.de/ocrocis/tutorial.pdf>. (zuletzt abgerufen am 17.06.2018).
- Springmann, U. und Lüdeling, A. (2016). OCR of historical printings with an application to building diachronic corpora: A case study using the RIDGES herbal corpus. *arXiv preprint arXiv:1608.02153*.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., und Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.
- Su, B. und Lu, S. (2014). Accurate scene text recognition based on recurrent neural network. In *Asian Conference on Computer Vision*, pages 35–48. Springer.
- Sudholt, S. und Fink, G. A. (2016). PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 277–282. IEEE.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., und Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

- TensorFlow Go (2018). https://www.tensorflow.org/install/install_go. (zuletzt abgerufen am 22.06.2018).
- TensorFlow Java (2018). Installing TensorFlow for Java. https://www.tensorflow.org/install/install_java. (zuletzt abgerufen am 2.06.2018).
- Tesseract 4.0 Training (2018). How to use the tools provided to train Tesseract 4.00. <https://github.com/tesseract-ocr/tesseract/wiki/TrainingTesseract-4.00>. (zuletzt abgerufen am 17.06.2018).
- Tesseract GitHub Repository (2018). `tesseract-ocr/tesseract`: Tesseract Open Source OCR Engine (main repository). <https://github.com/tesseract-ocr/tesseract>. (zuletzt abgerufen am 17.06.2018).
- Tombre, K., Tabbone, S., Pélissier, L., Lamiroy, B., und Dosch, P. (2002). Text/Graphics Separation Revisited. In *Proceedings of the 5th International Workshop on Document Analysis Systems V*, DAS '02, pages 200–211, London, UK. Springer-Verlag.
- Transcribus How-To (2018). How To use Transkribus – in 10 steps (or less). https://transkribus.eu/wiki/images/7/77/How_to_use_TRANSKRIBUS_-_10_steps.pdf. (zuletzt abgerufen am 17.06.2018).
- UB Bern Kartensammlung (2018). Kartensammlungen - Universität Bern. http://www.unibe.ch/universitaet/dienstleistungen/universitaetsbibliothek/recherche/sondersammlungen/kartensammlungen/index_ger.html. (zuletzt abgerufen am 04.06.2018).
- UB Bern Rhyner (2018). Details zur Sammlung Ryhiner - Universität Bern. http://www.unibe.ch/universitaet/dienstleistungen/universitaetsbibliothek/recherche/sondersammlungen/kartensammlungen/details_zur_sammlung_ryhiner/index_ger.html. (zuletzt abgerufen am 04.06.2018).
- van den Broecke, M. (2018). Cartographica Neerlandica Topographical names for Ortelius Map No. 60. <http://www.orteliusmaps.com/topnames/ort60.html>. (zuletzt abgerufen am 02.07.2018).

Literaturverzeichnis

- Veit, A., Matera, T., Neumann, L., Matas, J., und Belongie, S. (2016). Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.
- Virtuelles Kartenforum (2018). Virtuelles Kartenforum 2.0. <https://kartenforum.slub-dresden.de/>. (zuletzt abgerufen am 04.06.2018).
- Virtuelles Kartenforum Projekt (2018). SLUB Dresden: Virtuelles Kartenforum 2.0. <https://www.slub-dresden.de/ueber-uns/projekte/juengst-abgeschlossene-projekte/virtuelles-kartenforum/>. (zuletzt abgerufen am 04.06.2018).
- von Raumer, R. (1863). *Gesammelte sprachwissenschaftliche Schriften*. Heyder & Zimmer.
- Wahl, F. M., Wong, K. Y., und Casey, R. G. (1982). Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20(4):375 – 390.
- Wang, W., Yang, N., Zhang, Y., Wang, F., Cao, T., und Eklund, P. (2016). A review of road extraction from remote sensing images. *Journal of Traffic and Transportation Engineering (English Edition)*, 3(3):271 – 282.
- Yao, C., Bai, X., Liu, W., Ma, Y., und Tu, Z. (2012). Detecting texts of arbitrary orientations in natural images. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1083 –1090.
- Yu, C., Song, Y., und Zhang, Y. (2016). Scene text localization using edge analysis and feature pool. *Neurocomputing*, 175:652–661.
- Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., und Liang, J. (2017). EAST: an efficient and accurate scene text detector. *arXiv preprint arXiv:1704.03155*.
- Zitova, B. und Flusser, J. (2003). Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000.