



Cartography M.Sc.

Master thesis

Toponym Extraction from Selected Historic Topographic Maps Using Deep Learning

Gongmingyue Tang



2022

Statement of Authorship

Herewith I declare that I am the sole author of the submitted Master's thesis entitled:

"Toponym Extraction from Selected Historic Topographic Maps Using Deep Learning"

I have fully referenced the ideas and work of others, whether published or unpublished. Literal or analogous citations are clearly marked as such.

Dresden, 17.09.2022

Gongmingyue Tang

Acknowledgements

First of all, I would like to express my gratitude to the historical map semantic team: to my first supervisor Dr. Nikolas Prechtel for always being patient and ready to help. His passions of cartography and programming is a great inspiration for a python beginner like me; to my second supervisor Prof. Dr. Markus Wacker, for the detailed guidance towards a structured scientific way of academic work and thesis writing; to Jens Friedrich for his continuous help in technical questions.

A special thank goes to my cartography family. Before I started this master's study, I couldn't imagine that I was so lucky to meet such a lovely and warm group of people.

I also want to express my sincere appreciation to all the Stackoverflow and Github contributors. Without all the experience those gorgeous people shared, I would have been stuck in error message loops. My gratitude should also send to all the open-source projects and their detailed documentation, which helps a total beginner of deep learning to get on the right path.

Abstract

Historical maps contain a wealth of valuable information and are widely adopted as research sources in various fields. There is a need to find a proper approach to automatically unlock the information from a large number of digital map images. This thesis aims to contribute to the transformation of historical map scans into structured geo-data. The author tested a U-Net based pipeline on automatically extracting and recognizing the toponyms from the topographic map sheets from the Saxonian Land Survey (1780 – 1825). The thesis computed a method to generate annotated map patches that matches the unique style of map series, and applied it to train the detection model. The results show that generating synthetic data for training text detection models can be an effective solution to data scarcity in deep learning. With the proposed deep learning pipeline, the place names can be extracted with satisfactory accuracy. However, the recognition model still needs to be improved.

Keywords: Deep Learning; OCR; Historical Map; Text Detection; Text Recognition

Contents

Symbols and Acronyms	xi
1 Introduction	1
1.1 Motivation and Problem Statement	1
1.2 Research Objectives and Questions	3
1.3 Structure of the Work	4
2 Theoretical Framework and Literature Review	5
2.1 Theoretical Framework	5
2.1.1 Deep Learning	5
2.1.2 Optical Character Recognition	8
2.1.3 Synthetic Data	10
2.2 Literature Review	11
2.2.1 Early Work of Toponym Recognition from Maps	11
2.2.2 State-of-art Deep Learning based OCR	12
2.2.3 Recent Work of Toponym Extraction from Maps using Deep Learning	14
3 Material	16
3.1 History and Characteristics of Saxonian Milesheet	16
3.2 Toponym Hierarchy and Font	19
3.3 Virtual Map Forum 2.0	22
3.4 Pre-processing	22
4 Methodology	25
4.1 Workflow	25
4.2 Data Preparation	25
4.2.1 Training Dataset	25
4.2.2 Testing Dataset	28
4.3 Toponym Extraction	28
4.3.1 Trained Models from Existing Framework	28
4.3.2 U-Net Architecture	30
4.3.3 Performance Tuning	32
4.3.4 Post-processing	35
4.3.5 Word Box Localization	35
4.3.6 Evaluation Metrics	36

4.4	Toponym Recognition	37
4.4.1	Training Datasets	37
4.4.2	Deep Learning-based Text Recognition Tools	38
4.4.3	Evaluation Metrics	38
5	Results and Discussions	40
5.1	Text Extraction	40
5.1.1	Pre-trained Models	40
5.1.2	U-NET Pipeline	42
5.1.3	Evaluation	47
5.2	Text Recognition	48
5.2.1	Models Performance	48
5.2.2	Error Analysis	49
5.3	Overall Performance	50
6	Conclusion and Further Work	53
	References	55

List of Figures

1.1	Examples of some place names from one of the map sheets from Saxony, text were labelled in blue bounding boxes.	2
1.2	Deep Learning and Cartography relevant papers are shown in years, queried from Scopus.	3
2.1	Mathematical representation of a simplest artificial neuron is shown, with the input (x_m), weight (w_m), bias (b), summation function (\sum), activation function (φ), and output (y). (Rosenblatt, 1958)	6
2.2	Stochastic gradient descent compared with gradient descent. (Carpenter, Cohen, Jarrell, & Huang, 2018)	7
2.3	An example of a CNN architecture with two convolution layers in a feature extractor (Alom et al., 2018)	7
2.4	Illustration of downsampling operators	8
2.5	Components of a typical OCR system.	9
3.1	Reconstructed Reference System of the Berlin copy of Saxonian Map sheets. Figure from Müller (2018a)	17
3.2	Sheet System of Saxon Mile Sheets. Figure from Brunner (2005)	18
3.3	Examples of different levels of toponyms on Saxon Mile Sheets.	19
3.4	Examples of toponym placement on Saxon Mile Sheets.	20
3.5	Extracted Font Samples generated with the help of a template from Calligraphr	20
3.6	A set of ambiguous characters are depicted. (a) Letters that look similar (b) two different written letter "s", and (c) an example of the word "Koßwig", which challenges the text recognition	21
3.7	User interface of virtual map forum 2.0	23
3.8	Parts of two neighbored map sheets from Saxon Mile Sheets. The left one was mapped in 1803, and the right side is one from 1784. The place name "Groß Naundorf" was annotated on both parts but in different sizes and styles.	23
3.9	An example of an original map sheet downloaded from the map forum (left) and after re-projection and colour homogenization (right).	24
4.1	General Processing Workflow	26
4.2	Illustration of the data preparation procedure. The size of the images is not proportionally scaled.	27

4.3	Patches extracted from maps were classified into different feature groups (see section 3.4)	28
4.4	A set of cropped text areas use for model testing.	29
4.5	The architecture of implemented U-Net	30
4.6	Model performance tracked and visualised on TensorBoard	32
4.7	Comparison of model performance with different loss functions.	35
4.8	Text extracted and after enhancement	36
5.1	Detection Results using Pre-trained Models. Only problematic regions of test images are shown due to space limitation.	41
5.2	Predictions before and after improvement of training data	42
5.3	Predictions from models trained with different loss functions (BCE and BCE+0.2*IoU).	43
5.4	Predictions from models trained with synthetic data automatically generated and trained with manually labelled true map patches.	44
5.5	Prediction trained with manually labelled data	44
5.6	Predictions from models trained with synthetic data automatically generated and trained with manually labelled true map patches.	45
5.7	Detection output for map "Neustadt".	46
5.8	Detected small toponyms on map "Fuchschain"	48
5.9	Learning curve of the recognition model trained with synthetic data and validated with toponym patches. The character error rate decrease in runs but the gap between training and validation set is so large.	49
5.10	Confusion matrix for character recognition. Each row is a frequency count of characters recognized by the system, with darker colors associated with higher frequency counts. The matrix is case-insensitive.	50
5.11	District "Klein Pestitz" on historical map outlined in green and today's location on OpenStreetMap marked with blue box.	52

List of Tables

4.1	Time (s) to convolve 32x7x7x3 filter over random 100x100x100x3 images (batch x height x width x channel).	31
4.2	The training schedule of the recognition network	33
5.1	The evaluation of text box detection on different map sheets. Map names are the major cities in the mapping area of the pieces.	47
5.2	The performance of trained models on reconizing the toponym. The arrow in HTR+ column indicate the change of training dataset.	48
5.3	Part of the extraction and recognition result of map sheet "Dresden Neustadt"	51

Symbols and Acronyms

OCR	Optical Character Recognition	CRNN	CNN + RNN + CTC
DL	Deep Learning	GPU	Graphics Processing Unit
SGD	Stochastic Gradient Descent	BCE	Binary cross-entropy
ANNs	Artificial Neural Networks	IoU	Intersection over Union
CNNs	Convolutional Neural Networks	HTR	Handwritten Text Recognition
FCNs	Fully Convolutional Networks	CER	Character Error Rate
RNN	Recurrent Neural Network	WER	Word Error Rate
CTC	Connectionist Temporal Classification		

1 Introduction

1.1 Motivation and Problem Statement

Historical Maps are a treasure trove of valuable information about the past, and uncovering them has continuing relevance to studies nowadays. They expand the temporal extent for tracing back the dynamics of the earth, becoming a unique source for recording the earth's characteristics before modern earth observation techniques mature (Wu, Heitzler, & Hurni, 2022). Especially after the technological innovation of surveying in the 18th century, the precision of the national topographic maps became more reliable. Because of this, they are already used as references in broad academic domains. Researchers are adopting historical maps in reconstructing historic geographic features (Dunesme, Piégay, & Mustière, 2020; Wu et al., 2022), historical situations (Löki et al., 2020) and topographic patterns (Y.-Y. Chen et al., 2019; Ståhl & Weimann, 2022), man-made territories and borders (Uhl et al., 2021), political claims (Beatty, 2021), and many more fields of interest (Chiang, Leyk, & Knoblock, 2014; Reckziegel, Wrisley, Hixson, & Jänicke, 2021).

Despite the widespread digitization of historical documents, there is still a gap between scanned copies and the applicable data for modern research. A large number of scanned historical maps in high resolution are now publicly accessible on many platforms, such as Virtual Map Forum 2.0 (German: Virtuelle Kartenforum 2.0 der Sächsischen Landesbibliothek, <https://kartenforum.slub-dresden.de>), Historical Topographic Map Collection from USGS (<https://ngmdb.usgs.gov/topoview/>), Map of Switzerland (<https://map.geo.admin.ch/>). Although many of those maps are already geo-referenced, and the online portals allow the users to compare the historical maps with current landscapes visually, the interpretation still requires sufficient knowledge of cartography and history (Luft & Schiewe, 2021). This is especially when the legend is missing from the military mapping survey. The commonly adopted formats such as digital images, paper maps or Portable Document Format (PDF) of online copies of these historical maps can not be directly interpreted by computers (H. Li, Liu, & Zhou, 2018). Therefore, creating structured, tagged geo-data from scanned files is meaningful. Making the collections queryable and machine-readable will unlock the potential for further utilization of historical maps in modern analytic environments (e.g. GIS) (Chiang et al., 2014).

Text features, especially toponyms, are an essential component of maps. Jordan (Jordan, 2009) mentioned in the paper of four primary functions of place names on maps that they help map users to identify the location quickly, make the places queryable, enhance the understanding of geographical features, and indicate the change of local

1 Introduction

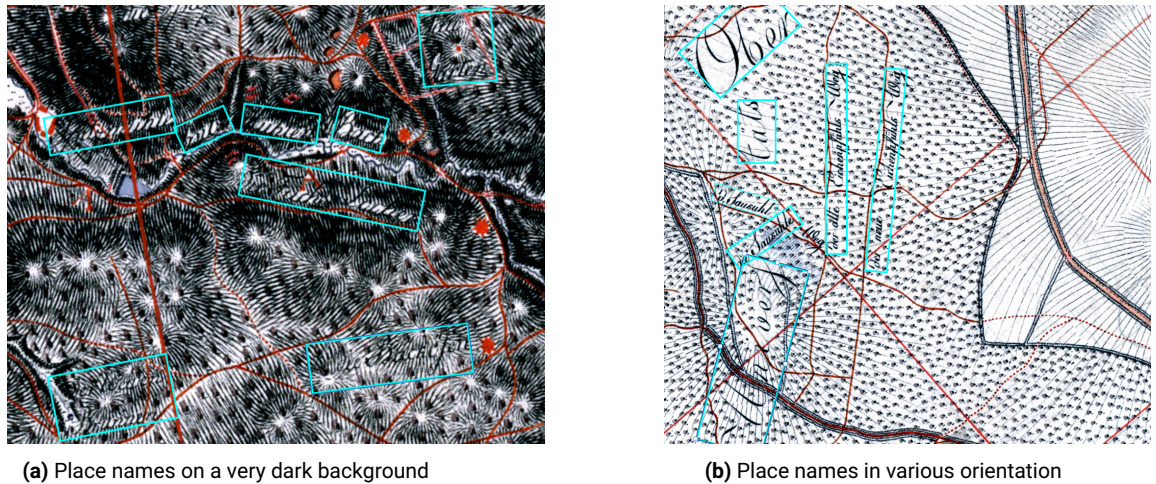


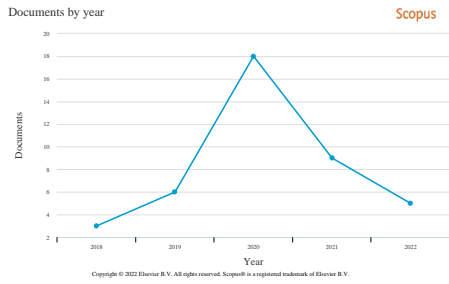
Figure 1.1: Examples of some place names from one of the map sheets from Saxony, text were labelled in blue bounding boxes.

culture and languages. In the information extraction process, the place names can be an entry point of map digitization (Barbaresi, 2018).

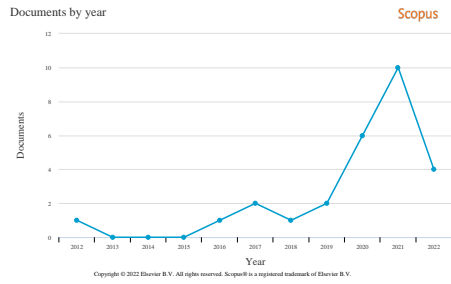
Manually labelling such a large volume of maps one by one is impractical. Together with the development of information technology, scientists have made a lot of efforts to automate the extraction of textual information from maps. Researchers have adopted traditional Optical Character Recognition (OCR) methods based on image processing algorithms for graphic segmentation (Chiang et al., 2014; Leyk & Boesch, 2010) and machine-learning using algorithms such as K-nearest neighbours and support vector machines to identify the characters (Pezeshk & Tutwiler, 2010, 2011). This has significantly improved the efficiency of digitization. However, for these approaches to achieve an optimal result, a good design of complicated algorithms and manual corrections are still needed. OCR is now well developed and can recognize modern text with very low error rates, but most tools are designed for text on simple backgrounds. The challenges for text detection and recognition in historical maps are unique. The text can appear in any orientation, in various sizes and unique typefaces, with different spacing and curvatures (see fig.1.1). The overlap of labels and other features, such as slope hachures, which are similar in colour, shape, or textures, makes separating even harder.

There is an increasing trend in using deep learning algorithms in historical map studies (Can & Kabadayi, 2021; Y. Chen et al., 2021a, 2021b; Laumer, Gümgümcü, Heitzler, & Hurni, 2020; H. Li et al., 2018). These attempts have optimized the automated information retrieving process by taking advantage of advanced computer vision techniques in scene text detection and mature text recognition applications. Achieving accurate

1.2 Research Objectives and Questions



(a) Papers contain "deep learning" and "cartography" in title-abstract-keyword



(b) Papers contain "deep learning" and "historical maps" in title-abstract-keyword

Figure 1.2: Deep Learning and Cartography relevant papers are shown in years, queried from Scopus.

generalization of predictive models requires sufficient training data (Lones, 2021). Although many high-quality labelled training datasets are now available for training and testing scene text detection models, their image distributions are clearly very different from the textual information on historical maps. Deep learning for map text detection still suffers from data scarcity. The cartographers tried to overcome the large dataset needed by, on the one hand, making use of existing pre-trained models (Can & Kabadayi, 2021; Laumer et al., 2020; Milleville, Verstockt, & Van De Weghe, 2020), on the other hand, artificially manufacturing synthetic data for training (Z. Li, Guan, Yu, Chiang, & Knoblock, 2021; Schlegel, 2021; Weinman et al., 2019).

Deep learning is still a new paradigm in the cartographic domain (see fig.1.2a), even less in historical map studies (see fig. 1.2b). Despite the inspiring work in recent years mentioned above on using deep learning in toponym extraction and recognition (see also section 2.2), a scientific paper is missing to look back at these efforts. Moreover, handwritten place names on more complicated and dated historical map documents were not tested. Therefore, the author would review the existing work and propose a deep learning pipeline based on the relevant framework. The pipeline would experiment on the map sheets from the Saxonian Topographic Survey (1780 – 1825).

1.2 Research Objectives and Questions

This thesis aims to contribute to research in the historic map semantics project. The overall research objective is to develop a pipeline enabling a step-by-step automated transformation of scans into collections of structured, tagged geo-data. This thesis will be mainly focused on toponym extraction and recognition and has the following sub-objectives:

1 Introduction

1. To explore the methods of synthetic training data generation and to compare the performance using synthetic and real data
2. To review existing deep learning architectures applied in text extraction and recognition and compare the performance of state-of-art pre-trained models on selected map sheets.
3. To develop an end-to-end working example from toponym extraction to text recognition on selected map sheets
4. To evaluate the performance of adapted models

The process would be divided into three main steps to meet the goal: For each part following questions need to be addressed:

- RQ1: Separation of text and graphic
 1. What are the existing deep learning pipelines for text extraction?
 2. Can deep learning on synthetic data achieve compelling results on real-world data?
 3. How well can toponyms be separated from the background?
- RQ2:Text recognition
 1. What are the existing text recognition methods?
 2. How well is the performance of the adapted text recognizer?
- RQ3: Evaluation
 1. How is the overall performance of the deep learning pipeline in toponym detection and recognition?

1.3 Structure of the Work

In chapter 2, the author first introduced the theoretical background of the thesis, writing about the OCR, deep learning, and semantic segmentation concepts. In the second part, the development of text detection and recognition in natural scenes and maps would be reviewed to answer research questions 1.1 and 2.1. In chapter 3, the writers talked about the reasons for choosing maps of the topographic survey of Saxony and an analysis of the type, placement and font characteristics of toponyms in this map series. Based on that, the author described the overall workflow of proposed pipeline and the methods used in each step. In chapter 5, the results were presented and discussed. In the last chapter, the author would conclude the work and the future direction.

2 Theoretical Framework and Literature Review

This chapter would cover the concepts and background of the work to offer a general understanding of fundamental theories. The author briefly introduced the terminology and definition of deep learning and image processing, including Semantic Segmentation and OCR. In the second section, the related studies in OCR would be outlined through the technology development timeline in order to answer the research questions about existing work about deep learning pipelines for text extraction and existing text recognition models. The focus would be on the applications, the performances and the shortcomings of the models.

2.1 Theoretical Framework

2.1.1 Deep Learning

As this thesis is on the implementation of a deep learning model in text extraction and recognition, the introduction of deep learning technology in this section would only focus on relevant background and terms. The author would briefly present the history, go through the advantage and challenges of applying deep learning, and then come to the basic neuron model and the architecture of the convolutional neural network.

The deep learning concept can be traced back to Walter Pitts and Warren McCulloch's work of proposing a computer model based on neural networks in 1943 (Pouyanfar et al., 2019). The main concept behind this was to mimic the structure of human brain. From the 2000s, deep learning demonstrated a broad and profound impact in many fields, proving its powerful predictive potential for data (Pouyanfar et al., 2019; Sarker, 2021). But in the intervening seven decades, it has experienced lows and been questioned many times. The slow but steady evolvement of deep learning to its current role mainly benefited from the introduction of backpropagation by Paul Werbo, exploration of models and architecture from shallow neural networks and deep neural network, and important development of computer hardware, which support the processing of big data (Sarker, 2021; Wang & Raj, 2017). Deep learning's recent success is due less to its earliest ambition, which was to simulate the human brain and more to its deep architecture (Wang & Raj, 2017).

Many articles have already discussed the reasons deep learning became so popular (Pouyanfar et al., 2019; Sarker, 2021; Wang & Raj, 2017). The researchers figured universal learning approach, robustness, and generalisation are the main advantages of its ability

2 Theoretical Framework and Literature Review

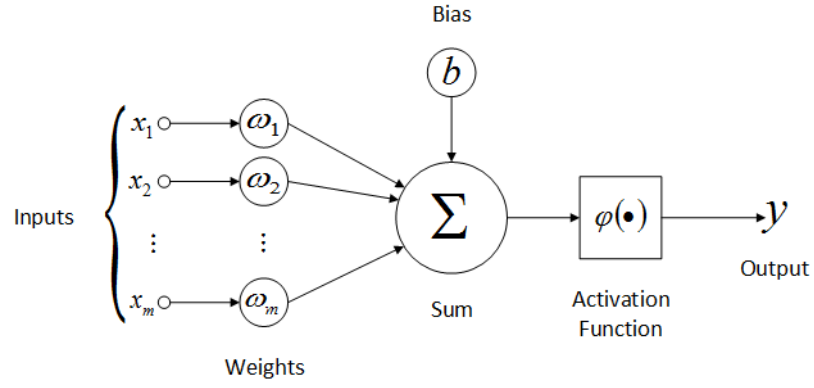


Figure 2.1: Mathematical representation of a simplest artificial neuron is shown, with the input (x_m), weight (w_m), bias (b), summation function (Σ), activation function (φ), and output (y). (Rosenblatt, 1958)

to learn features on different tasks in various domains (Alom et al., 2018). However, the extensive training data needed, hardware dependencies, long training time and interpretability challenge the application, which needs to be taken into consideration before use (Sarker, 2021).

Figure 2.1 shows the function of a simple artificial neuron of deep learning. The mathematical modeling denoted the input data as a vector $\chi = \{x_1, x_2, \dots, x_m\}$, and designated weight vector as $W = \{w_1, w_2, \dots, w_m\}$. An adder would sum the input, weighted by the weight vector. The operations described here constitute a linear combiner (Sánchez-Ramírez, Segovia-Hernández, & Hernández-Vargas, 2020). A real number called bias (b) can be added to the preliminary outputs of the linear combiner to apply the affine transformation. An activation function (φ) would then be adopted to convert the weighted input of the neuron to the output. The main goal of deep learning is to find the optimal weight elements by training. The best weight vector should produce the results which have the least error or loss between the actual and the predicted values across different samples in the data set (Sewak, 2019). This loss is calculated by a chosen loss function.

Gradient descent approach is used in deep learning to minimise the loss function. This first-order optimisation algorithm to find the local minima has the drawback of a long processing time. Therefore, the deep neural networks are widely trained with the Back-Propagation algorithm with Stochastic Gradient Descent (SGD) (Rumelhart, Hinton, & Williams, 1986). The SGD, compared to the classic gradient decent algorithm, only calculate for a single, randomly selected point at each step (see fig. 2.2), which increases the efficiency.

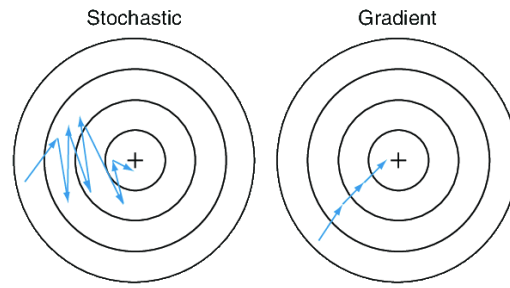


Figure 2.2: Stochastic gradient descent compared with gradient descent. (Carpenter et al., 2018)

Deep learning systems are usually classified into three categories based on the amount of labelled data they use: supervised, semi-supervised or unsupervised (Sarker, 2021). While there is a trend of developing unsupervised and semisupervised models, for example, Generative Adversarial Networks (GAN), in order to reduce the human work of labelling training data, the majority of the existing deep learning implementations are supervised algorithms (Wang & Raj, 2017). For image analysis tasks, Convolutional Neural Networks (CNN) are commonly employed (Sarker, 2021).

Figure 2.3 visualised a CNN architecture. A convolution neural network comprises three types of hidden layers positioned between the input layer and the output layer. These are convolutional layers, pooling layers, and fully-connected layers (shown as classification in the figure). A CNN network may contain multiple instances of these kinds of layers sequentially arranged in a particular order, with the convolutional and pooling layers primarily interleaved before all fully connected layers (Sewak, 2019). The last fully connected layer drives the final classification decision.

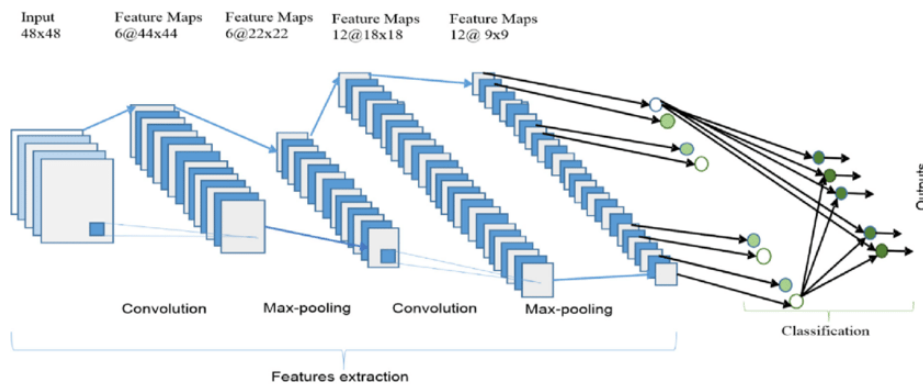


Figure 2.3: An example of a CNN architecture with two convolution layers in a feature extractor (Alom et al., 2018)

2 Theoretical Framework and Literature Review

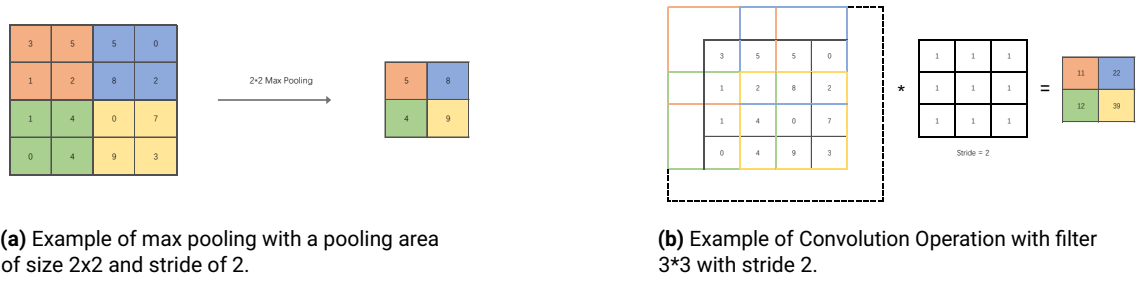


Figure 2.4: Illustration of downsampling operators

"Convolutional" in the term CNN comes from the convolution layer. Each convolution layer will produce a set of feature maps generated from each band of the input convolved with a kernel filter. The pooling layer, also known as the downsampling layer, compresses the feature maps passed from the convolution layer. Its primary purposes are first, to reduce the number of features, thus reducing the number of parameters and simplifying the complexity of the convolutional network computation and secondly, to achieve translation invariance (Yamashita, Nishio, Do, & Togashi, 2018). While a majority of applications use either average-pooling or max-pooling layer, the team of Springenberg (2014) proposed a new architecture, which removed all pooling layers. To do subsampling, they used convolution with strides. This convolution net was experimented with and proved to be concise and with good performance. As it is shown in figure (2.4), the convolution filter with strides and max pooling can both reduce the resolution of the in-between representation of networks. But the convolution can keep learning the properties while pooling reduces the complexity.

2.1.2 Optical Character Recognition

Optical Character Recognition (OCR) is defined as the process of converting optical patterns contained in a digital image into its constituent characters (Chaudhuri, Mandaviya, Badelia, & Ghosh, 2017; Islam, Islam, & Noor, 2017). The history of character recognition is even longer than the invention of the computer (Islam et al., 2017). The earliest OCR appeared as a mechanical device. Its development in the last eight decades can be classified into four generations (Berchmans & Kumar, 2014). Only a few typefaces and character forms can be recognised by the first generation. The second generation can transcribe both handwriting and printed text, but only numerical, few letters and symbols. The third generation has been created with significant advancements in hardware technology. They work with a larger set of handwritten characters than before and characters with low print quality. The fourth generation of OCRs can scan and read characters from complicated documents that are mixed in with text, tables, and mathematical symbols, as well as low-quality noisy documents, free-form handwriting,

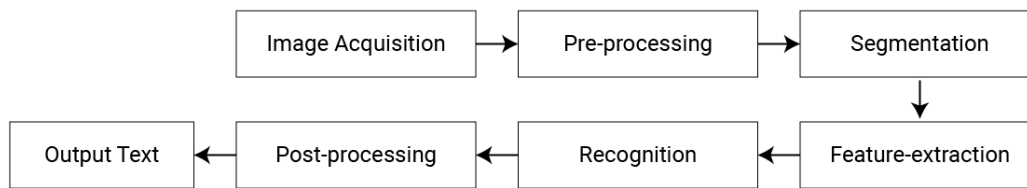


Figure 2.5: Components of a typical OCR system.

and colour documents. From its starting point, the efficiency, robustness and versatility have largely improved.

A classical OCR system is composed of six phases, as shown in figure 2.5.

The initial step of OCR is image acquisition, which means the digitisation of documents with the help of an electronic device. The dataset input affects the recognition accuracy significantly (Islam et al., 2017). Despite the mature scanning technology today, which produces near-lossless image coding, the distortion, low accuracy and noise problem existing in images themselves, especially on historical and handwritten documents, pose challenges for subsequent recognition steps.

Pre-processing is therefore one of the most important OCR procedures. The sequence of adjustments made to the scanned images improves the image quality and has a direct impact on the effectiveness and dependability of the next steps of recognition. The most commonly adopted method is thresholding which binarises images (Lázaro, Martín, Arias, Astarloa, & Cuadrado, 2010). Different types of filters, such as mean, and median filters, are employed for noise removal. In some cases, various morphological operators, including erosion, dilatation, opening, and shutting, are applied (Serra & Vincent, 1992). Additionally, normalisation techniques are used to produce pictures that are uniform in size and shape for segmentation and feature extraction (Avinash, 2018).

Segmentation comes next in the OCR process. Segmentation describes the process of breaking the whole image into subparts to identify what is actually comprised in the input image (Islam et al., 2017). Sequentially the document is divided into lines, then to the word level, and finally to the character level to reduce the complexity and increase the efficiency of further processing. However, as it's a key step of the classic OCR system, incorrect word segmentation causes problems. Segmentation is thought to be the cause of around half of character recognition mistakes (C.-H. Chen & DeCurtins, 1993). Therefore, segmentation-free systems were proposed and tested to be a promising alternative (C.-H. Chen & DeCurtins, 1993; Sabbour & Shafait, 2013).

2 Theoretical Framework and Literature Review

For each segmented character, its various features are then extracted. An important research topic is how to choose the appropriate characteristics and how many features should be employed overall (Islam et al., 2017). The features extracted for their different representation and purposes can be categorised into structural features, statistical features, and global transformations (Sharma, Kaushik, & Gondhi, 2020).

Based on the feature retrieved in the previous stages, the characters will be classified into the appropriate categories with the help of different classifiers. In the early period, pattern recognition adopted template matching with a library of certain predefined templates for each character of each font (Chaudhuri et al., 2017). Artificial neural networks (ANNs), kernel methods, statistical techniques, and structural pattern techniques are currently employed as other general recognition approaches (Memon, Sami, Khan, & Uddin, 2020). These methods aren't always distinct from one another or independent of one another. On occasion, an OCR method in one approach may also be regarded as a component of another approach (Chaudhuri et al., 2017).

Post-processing is a process to increase the accuracy of recognition results. Two directions are, respectively, using multiple classifiers or conducting contextual analysis (Islam et al., 2017). The probability of character sequences in words can be a hint to correct errors. More directly, fuzzy searching in dictionaries helps to identify errors and transfer to the most similar word (Chaudhuri et al., 2017).

The analysis (Memon et al., 2020) of OCR research between 2017 and 2019 shows that the deep learning approach is a trend. Improved classification accuracy was achieved using deep learning, but the cost was computing complexity, especially during the training phase. Widely applied machine learning and deep learning architectures are Support Vector Machine, Artificial Neural Networks, Naive Bayes and Convolutional Neural Networks (Sharma et al., 2020). Among these, convolutional neural networks are one of the most popular (Simonyan & Zisserman, 2014) and help to achieve unprecedented accuracy (Meduri & Goyal, 2018). In section 2.2.2, a few state-of-art architectures would be reviewed.

2.1.3 Synthetic Data

The use of synthetic data started together with the history of computer vision (Nikolenko, 2021). Having a significant amount of training data is important to help the computer vision algorithms to get a good performance (Gupta, Vedaldi, & Zisserman, 2016). The preparation of manually labelled datasets for each one of the learning tasks is time-consuming. Synthetic datasets provide an affordable and scalable alternative to manually annotating (Gupta et al., 2016). The benefits of synthetic data generation and enhancement techniques are twofold. On the one hand, it facilitates the acquisition of

more data from a limited number of data sources, and on the other hand, it helps to suppress overfitting errors (Khosla & Saini, 2020).

Artificially inserting text instances in appropriate semantic locations in a neutral image background can produce large-scale comprehensive annotation for training (Khan, Sarkar, & Mollah, 2021). The main principle is to render the proper font and colour on background images. Shadow, noise and projective distortion are introduced to the image to make the imitation more realistic (Grond, Brink, & Herbst, 2016).

Augmentation is another solution for data deficits. Data augmentation describes a series of transforming techniques that are utilized to increase the quality and size of images (Khosla & Saini, 2020). Based on data warping, common operations are geometric transformations such as affine transformations, cropping, rotating, colour space transformations such as RGB to HSV, and random erasing.

2.2 Literature Review

2.2.1 Early Work of Toponym Recognition from Maps

Morphological operators are one of the most popular image processing techniques, which was also one of the starting points for the semantic segmentation of historical maps. Different mathematical morphological operations have also been widely adopted in text separation from maps. Yamada (1993) extracted text and symbols on topographic maps by employing directed feature planes for erosion-dilation processes, which is computationally intensive. Based on this, Biswas and Das (2012) tested a procedure of sequentially employing horizontal and vertical mathematical morphological operations to line the text to an entity, combining with mean and standard deviations to separate text elements. The authors also compared the method with text extraction based on intensity analysis and pointed out that the morphology-based method performs better under specific language contexts. These works, however, do not apply to the case of text with multi-angle.

The connected component labelling method was also one of the early efforts. Geometric characteristics such as height, width, or area of the text were investigated to be distinguished from other line features (Chiang et al., 2014). An example is one of the very early works of Li et al. (2000) to locate and recognise street labels. The label boxes were extracted by character alignment and size, together with the graphical background of the label positions adjacent to the street lines. For recognition, the researchers manually rotated the text to horizontal orientation and utilised a segmentation-free classifier to recognise the entire string directly. This recogniser may use overlapping line information and is tolerant of overlapping graphic segments. The obtained accuracy seems to be

good enough to use context-based analysis. However, even after intricately planned processing steps, many boxes still contain some graphical fragments, and some text characters are missed from the detection. However, the concept of feature extraction based on attributes-shape descriptors and pre-processing to improve recognition accuracy is valuable.

Other early segmentation attempts also included clustering analysis and the image pyramid method. Most of the earlier works were continuously improving based on these above-mentioned categories of approaches. Pezeshk and Tutwiler (2011) Markov models were operated to complete broken characters. Velázquez and Levachkine (2004) used four imaginary directional lines (V-line) that allow slightly curved text labels to be extracted. Pouderoux, Gonzato, Pereira, and Guitton (2007) added string analysis to the connected component algorithm for concatenation. Chiang and Knoblock (2011) employed morphology-based operators to determine the directional rotation of strings. Weinman (2013) used information from supplementary sources such as gazetteers to detect and correct errors. These steps are still valuable and effective image processing methods today. However, they cannot accomplish the tasks in more complex map contexts. They are not general solutions and require selecting appropriate structured elements for the corresponding maps to achieve the goal.

2.2.2 State-of-art Deep Learning based OCR

Deep learning-based OCR always consists of two steps: text detection and text recognition, i.e., the target image is input to the text detection algorithm to get a word box, and each bounding box is fed to the text recognition algorithm to get the recognition result.

Deep learning-based text detection algorithms are broadly divided into two categories: proposal-based regression algorithms and segmentation-based algorithms (Qin & Zhang, 2020). Additionally, to increase the detection accuracy, some studies also tried hybrid methods, which combined these two types of algorithms (Khan et al., 2021).

In the proposal-based technique, rectangular or quadrilateral text boxes are convolved in various directions over the whole image to identify text areas. This approach was initially derived from a target detection algorithm and then tailored to the properties of text boxes. CTPN (Connectionist Text Proposal Network) (Tian, Huang, He, He, & Qiao, 2016) and SegLink (Shi, Bai, & Belongie, 2017) are two representative methods of proposal-based methods.

Considering that the length of text boxes varies and text could exist in a long rectangle, CTPN (Tian et al., 2016) split the detection task into two steps. The author suggested first assessing whether a small part of the text belongs to a phrase and then merging

the small text boxes to get the entire string. As contextual information is always crucial for characters in the text, a Recurrent Neural Network (RNN) is used to enhance the prediction further. The shortcoming of CTPN is that it can not detect non-horizontal text. The SegLink (Shi et al., 2017) algorithm is similar to CTPN in that it also proposed to first detects small pieces of text lines and then merge them together. But the network structure adopts the single-shot detector (SSD), which detects text at multiple scales of the feature map and then fuses the results. In addition, the researchers added the regression of angular information to the output. However, SegLink is not working on the text with large spacing or curved. DRRG (Deep Relational Reasoning Graph) (Zhang et al., 2020) is another regression-based method which applies the concept of connected component analysis. DRRP can detect the text in any shape and orientation. This method treats positioned small rectangular components as nodes. It uses a graphical convolutional network to infer the relationship between the nodes and thus correctly connect the components to the text instance. The model was tested to perform well on multi-directional and multilingual scene text.

In segmentation-based text detection, segmenting network structures execute pixel-level semantic segmentation and use the segmented output to build text lines. A representative algorithm is TextSnake (Long et al., 2018). The authors provided a flexible way of representing text lines. The Fully Convolutional Network (FCN) is first employed to predict text line centerlines, and to infer text line regions and their geometric properties. The masked text centerlines help to obtain the feature map. Based on this, the word instance can be separated with the help of the disjoint set. This method is the first effective model to detect curved text. Traditional post-processing methods for segmentation-based text detection are complicated. Therefore, DBNet proposed to add differential binarization into the segmentation nets. The additional binarising step helps simplify the post-processing and enhance the performance of text detection.

FCE (Zhu et al., 2021) is a very new text detection method that combined regression and segmentation. The study suggested using regression to convert each pixel in text instance to Fourier feature vectors. On the segmentation side, the mask of the text region is predicted. The vectors are then embedded to mask the area through Fourier Transformation. FCENet can effectively detect arbitrarily shaped text and proved to have excellent generalization ability.

The deep learning-based text recognition algorithms have a relatively unified principle (Qin & Zhang, 2020). The most dormant structure of the recognition framework consists of CNN+RNN+CTC, which is commonly known as the CRNN structure. CTC in the framework refers to the Connectionist Temporal Classification, which functions as a step to find the best path decoding. The structure is proved to be stable and robust in transcribing words. So most of the OCR software (for example, EasyOCR, KerasOCR,

and PaddleOCR) adopts this structure and makes improvements to increase efficiency and save computational memory.

The CITlab team at the University of Rostock developed a Handwritten Text Recognition engine (HTR) based on TensorFlow (Michael, Weidemann, & Labahn, 2020). Bidirectional long short-term memorys (BLSTMs) are applied as RNN layers to enhance context modelling. Downsampling is added to reduce the model size, which increases the processing speed and keeps the performance. The model CRNN-STN in the MMOCR toolbox (Kuang et al., 2021) uses a spatial transformer network (STN) (Shi, Wang, Lyu, Yao, & Bai, 2016) as a preprocessor, which optimizes the skewed text and automatically detects skewed text boxes without special markup. Another direction of text recognition is using attention-based methods. The main focus of this method is on the correlation between the parts of the text sequences.

Another direction of text recognition is using attention-based methods. The main focus of this method is on the correlation between the parts of the text sequences. ABiNet (Fang, Xie, Wang, Mao, & Zhang, 2021) is one of the attempts. The researchers refine the recognition results by Language Model, making full use of the contextual semantic information. The BCN (Bidirectional cloze network) module is used there to predict or correct the currently recognized characters. This makes the model have better recognition ability under low-quality imaging conditions.

2.2.3 Recent Work of Toponym Extraction from Maps using Deep Learning

In recent years, cartographers have started to apply the achievements of robust reading in computer vision area to the field of historical maps. Scholars have experimented with different deep learning architectures on various types of map atlas in different languages from different regions worldwide. The proposed methods were tested on cases from cadastral maps to more complex topographic maps, from black-and-white to colour images, and from contemporary to more dated hand-drawn maps. Mature commercial and open source text recognition tools boosted these works. These studies show excellent findings and a promising future of using deep learning to unlock the potential of historical map resources.

The research from Li et al. (2018) proposed a state-of-the-art framework for map text and feature recognition. The reader applied deep learning for text detection on maps, followed by the separation via graph-based segmentation and clustering algorithms, an OCR engine to interpret the text labels, and the use of a gazetteer as the source to add contextual understanding. The outcomes confirmed the effectiveness of the proposed approach for map text recognition and map content comprehension. The framework demonstrated a promising structure for a deep learning-based OCR job on a map, even

though additional technique development is required for more complicated and dated map series.

The study of digital map processing benefits from the computer vision domain. Weinman et al. (2019) proved in the paper that deep neural networks, which were originally not designed for cartography, can be adapted and used on maps. The authors developed a map text detection network from a CNN and an RNN framework to detect and recognise the labels on USGS historical maps. A map text synthesiser was applied to generate unlimited training data without overfitting. The paper provided insight into tailoring non-domain-specific deep learning systems for automating map digitising.

Kartta Lab (Tavakkol et al., 2019) is an open-source and open-data project aiming to develop a framework to generate vectorised, machine-readable data from historical map images automatically. As a module of this framework, Z. Li et al. (2020) developed an end-to-end approach using a consensus model combining the results from a textual predictor and a visual predictor to extract location phrases.

Can and Kabadayi (2021) trained two CNN-based text detection models using pretrained Resnet-50 and UNet architecture and tested them on Austro-Hungarian historical military mapping survey. The results of this study show that the pre-trained Resnet-50 model has a better performance in detecting text areas from historical maps. The Python Tensorflow-based toolbox trained with IAM-database, however, has a 42% character error rate, which needs further improvement.

Schlegel (2021) made use of an existing text detector, Strabo, which was designed to detect and recognise the text on maps. The article suggested a series of adjustments to transfer the universal approach to individual map series, which largely improved the performance of the model. The F1 score was increased from 58% to 77%. However, the test case is relatively simple, without many interfering features in the background. The alignment of labels to the current context in OpenStreetMap and validation with the local gazetteers might only be feasible for modern street maps.

Laumer et al. (2020) used CNN to segment pixels into label and background. The single characters were then clustered into phrases with the help of DBSCAN. For interpretation, Google's Vision API was used, and a local gazetteer helped to refine the result. Problems occurred in the disparity of place names in different languages and time periods. The author didn't present a quantitative evaluation of pipeline performance.

3 Material

The target maps of this thesis are the selection sheets of the map series from the third major survey of Saxony from 1780 to 1825 at a scale of 1:12.000. The entire map series has no official map name, but for its units of measurement, they are widely called 'Saxon mile sheets' (German: Sächsische Meilenblätter).

The work of the Historical Map Semantics team at the Dresden University of Technology and the Dresden University of Applied Sciences is the starting point for this thesis. The team focused on the extraction of the polygon and line features from the Saxon Mile Sheet. The data preparation of the maps by Dr Nikolas Prechtel also provided a foundation for this thesis.

3.1 History and Characteristics of Saxonian Milesheet

The land survey was motivated by military needs. Accurate knowledge of the terrain was essential to the command of troops, and the changes in warfare that took place in the 18th century demanded topographical maps with very high accuracy (Stams & Stams, 1981). During the Seven Years' War, Saxony still lacked large, uniform maps inside its territory. The existing official maps no longer met the requirements, so for both military and post-war economic development purposes, the Saxon region and its entrances were required to be surveyed in detail. The topographical maps wanted were proposed to serve also for cameralistic purposes, especially as a reliable graphic basis for mining, roads and water engineering (Stams & Stams, 1981). As a result, the Saxon Milesheets are exceedingly detailed. All the terrain objects that could be of importance were surveyed. According to the records of Nagel (1876), the objects of interest are in the range of the complete network of roads from the roads to every country lane and the smallest footpath, the entire hydrographic network down to the smallest streams and drainage ditches, all forests, meadows, huts and ponds, and in the complex of the villages, as far as the scale allows, every single building with its yard and garden.

A prerequisite for using the historical map series for today's research topics is the correspondence of the coordinate reference systems. At the time of the Saxonian topographical land survey, the geodetic-mathematical knowledge, as well as the instruments and measuring methods, corresponded to the contemporary level and the state-of-the-art. Stams (1981) described the survey process in their paper. The engineering corps started practical surveying in the autumn of 1780. The first work was to secure a baseline on the flat area southwest of Pirna. The measurement of the main trigonometric network followed the marked endpoints of the base. In the

3.1 History and Characteristics of Saxonian Milesheet

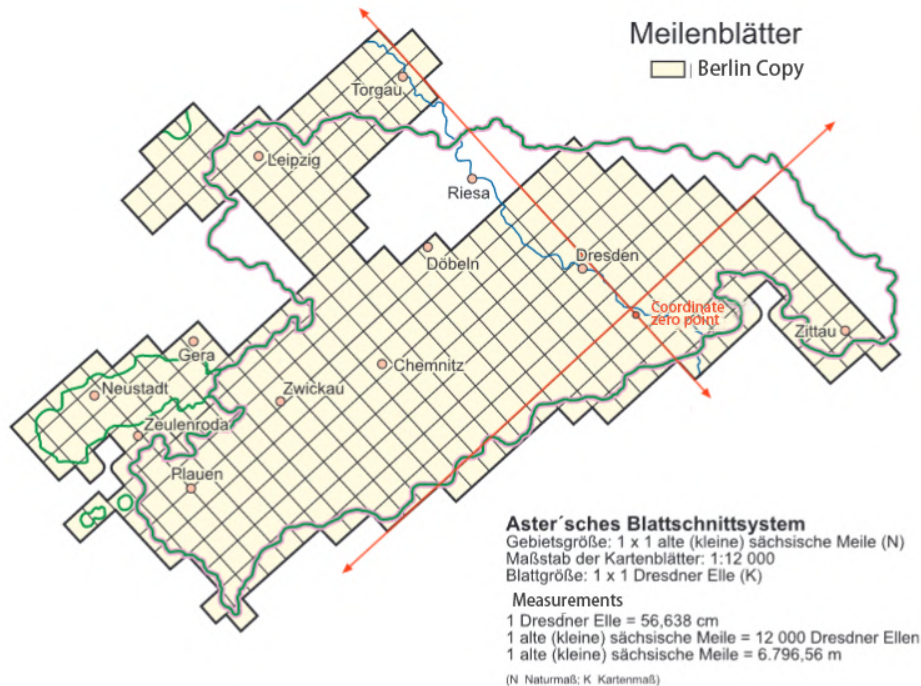


Figure 3.1: Reconstructed Reference System of the Berlin copy of Saxonian Map sheets. Figure from Müller (2018a)

beginning, neither an azimuth observation nor a geographic positioning was carried out in connection with the triangulation. Therefore the calculated triangulation points could only be referred to as the extended baseline in terms of coordinates (Stams & Stams, 1981). The lateral boundary of the individual miles forms an angle of 42° W with the north direction. The longitude measurements were not yet standardized, and the orientation to the north was missing at the time of the Saxon triangulation. It was only oriented much later by reference to the meridian of the Mathematical-Physical Salon in Dresden. Hans Brunner's work in the early 2000s helped to reconstruct the historical system. As shown in the figure 3.1, square map sheets are mosaicking and georeferencing without overlapping (Walz & Berger, 2003).

The lasting value of the mile sheets lies in the accurate documentation of the state of the country at the end of the 18th century. After the mid-18th century, the first series of topographic maps based on geodesy were produced in various European countries. This map series of Saxony occupies a top position in terms of scale as well as in terms of surveying and cartographic execution. The maps are, for the circumstances of the period, of high accuracy in their geometry. Their representations of the cultural

3 Material

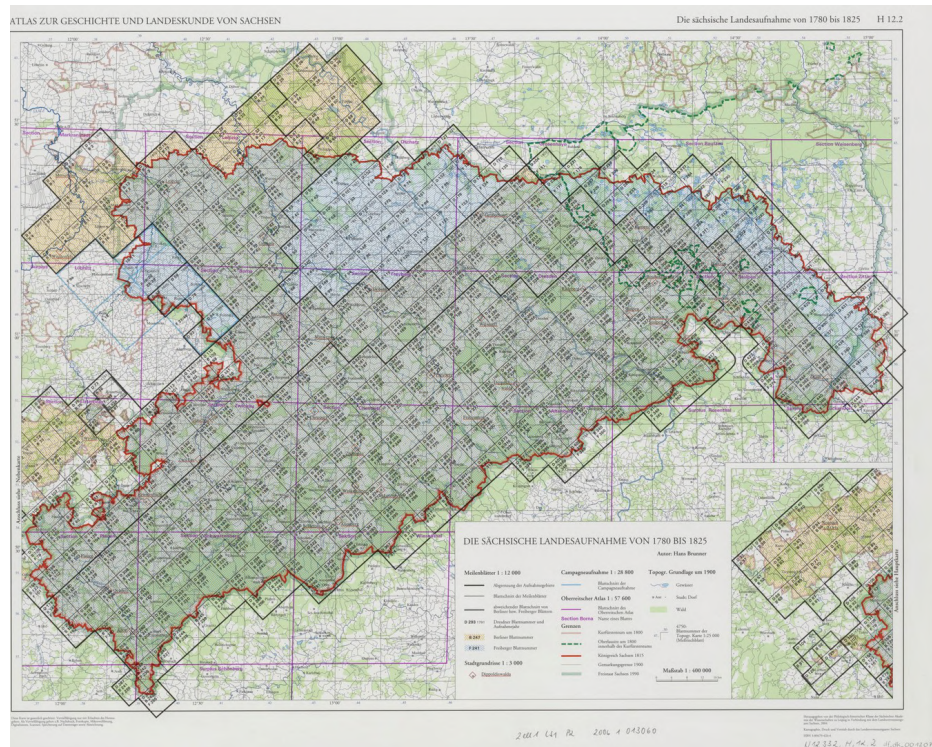


Figure 3.2: Sheet System of Saxon Mile Sheets. Figure from Brunner (2005)

environment come out to be a solid foundation for determining regional placement and modern cultural landscape features (Walz & Schumacher, 2011).

The dissertation used the Berlin copy of mile sheets. The sketching mapping was finished on quarter sheets and then glued together to make a mile sheet. Each quarter has information on the topographer and the sheet number written on the back (see fig.3.2). The completed sheet was kept by the General Staff and later became the so-called Dresden copy of the mile sheets, which was held in the King's cabinet and became today's Berlin copy. The figure shows the sheet system of the map copies (Brunner, 2005). The figure can also tell that map sheets of the northern Saxon territories are missing in the Berlin copy (square pieces with yellow slashes), which were surveyed between 1819 to 1825. They were added to the copy of Freiberg (square pieces with blue slashes). 180 Berlin mile sheets documented the contents of the original surveys and were better preserved than the Dresden copy. Therefore, in 2006, the Berlin copy served as the basis for digitization in cooperation with the Sächsische Landesbibliothek - Staats- und Universitätsbibliothek Dresden (SLUB) (Müller, 2018b).



Figure 3.3: Examples of different levels of toponyms on Saxon Mile Sheets.

3.2 Toponym Hierarchy and Font

As introduced in the last section, the Saxon mile sheets include a large number of details. All place and field names were well annotated and included in the copies. There can be more than a hundred toponyms marked on one map sheet. Accurate identification of the structure of the place names is of great help in improving the accuracy of the place name classifier (Adelfio & Samet, 2013). The place names in the Saxonian map series can be categorized into three levels, as shown in examples (see fig. 3.3). They are distinguished in terms of font size and form.

The three orthogonal dimensions of prominence, geographic containers, and feature types may be used to determine the categories of toponyms in hierarchies (Adelfio & Samet, 2013). The class of objects in this map series can be described by the feature type.

Cities and regions were labelled with the largest font size, all capitalized and on simple backgrounds without much distraction from other geographical features. The shapes are clear, and the strokes are mostly solid and in black or dark brown colour. The letters are widely spaced apart, which makes them may be well segmented in text recognition. They are all parallel to the mapping baseline, in other words, horizontally written on map sheets.

Towns, large villages, districts and rivers are annotated in medium-sized type and with thinner strokes. Place names are written next to settlements. The river names were often placed in the middle of the channel, aligned to the direction. The words are capitalized and are more handwritten, styled with thin to thick shades of strokes and interlocking letters. The transition of stroke thickness involved the variation of colour from visually black to light brown or grey. The classic character separation step in OCR cannot cut out individual letters from such cursive joined text.

3 Material

Buildings such as brickyards, mills, different types of roads, landscapes such as mountains, lawns, forests, ponds and so on are marked in the smallest font size. Text at this level can only be recognized when the users zoom in on the region. The names are written in any direction, especially for line features such as the footprint trails, where the text is aligned along the path. The font style also varies from the larger text in order to make the place names readable at this level of the font size.

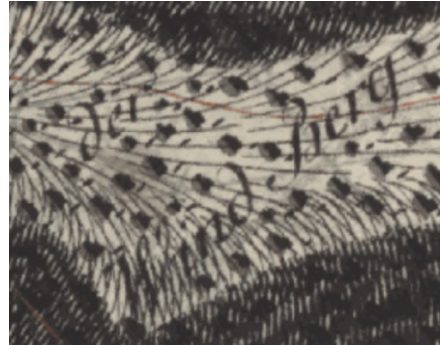


Figure 3.4: Examples of toponym placement on Saxon Mile Sheets.

The placement of the text at the second and third levels mostly consciously avoids intersection with the trees indicated by the dark dot feature but overlaps on linear features such as paths and hatch lines (see fig. 3.4). This makes traditional OCR methods to filter out non-textual features very difficult.

The text on the maps is in cursive handwriting, which challenges recognition. The handwritten fonts are unique, and it is impossible to find similar calligraphy in the existing font libraries. For example, the letters "s", "z", "ß", and many more are very



Figure 3.5: Extracted Font Samples generated with the help of a template from Calligraphr

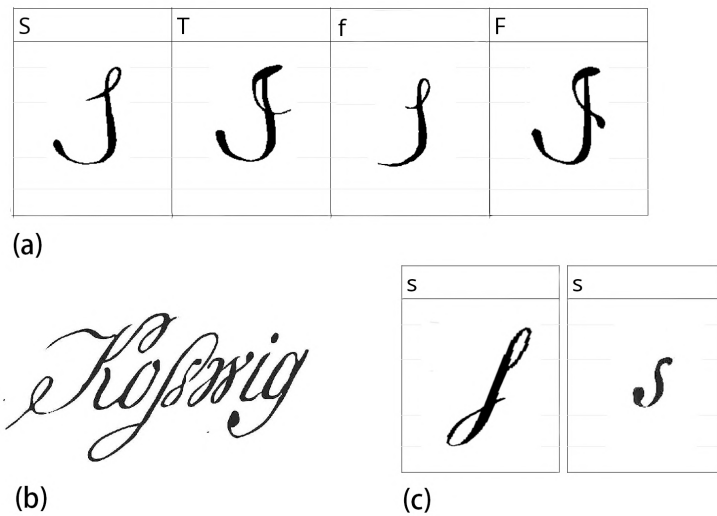


Figure 3.6: A set of ambiguous characters are depicted. (a) Letters that look similar (b) two different written letter "s", and (c) an example of the word "Koswig", which challenges the text recognition

different from the modern style (see fig. 3.4 and fig. 3.5). Also, since all the letters are handwritten, the characters are non-standardized. Therefore, to generate synthetic annotation for training, the author needs to first extract a collection of unified, simplified text from the map to simulate the handwritten font. Figure 3.5 shows the customized character template that mimics the antique font style of the Saxon Mile Sheets. The template was created by the Historic Map Semantic team. Based on these, the author generated the font file in TFF format with the help of an open-source font creation website - Calligraphr. Proper spacing and italic formatting were set in the font file after this.

The ambiguity associated with cursive handwritten text has always been a problematic part of text recognition. Figure 3.6a shows some of the characters, such as uppercases of "S", "T", and "J", which look similar to each other. The lowercase 's' is particularly special in the map font, which has more than one variant (see fig. 3.6b). The letter "ß" is usually written as "ss" with a combination of these two variants, but the sequence of variants is random. In this case, contextual analysis can be a correction to identify the actual result. For example, "sch" is a common letter combination in German, and "-dorf" is a common ending of village names on maps. This can be an additional factor to distinguish lowercase "s" and "f".

Because historical maps have been preserved over a long period of time, places written in smaller font sizes and finer and lighter colours are challenging to separate from the background. The names of streets, routes, and minor terrains have changed significantly

3 Material

over time and are relatively secondary for modern research topics and for generating machine-readable maps. Therefore, this paper only focuses on the largest and second-largest font sizes, which are almost always parallel to the map slices and does not extract smaller texts.

3.3 Virtual Map Forum 2.0

The Map Forum of the Saxon State Library - The Saxon State and University Library Dresden (German: Sächsische Landesbibliothek, abbr. SLUB) offers digital access to over 20,000 historical maps. The early DFG project has helped to digitize these widely covered cartographic masterpieces in high resolution (BILL, WALTER, & MENDT, 2014). With the development of digital science, spatial and temporal research topics set higher requirements for historical map service. First and foremost is the need to transfer the digitized material into a spatial context with georeferencing. Therefore, to further open the public access to the data and provide the possibility for the researchers to download and integrate the data into their local working environments, in April 2013, the SLUB initiated the project "Virtual Map Forum 2.0" (Mendt, 2014).

A visualization within georeferencing deals with the translation of data into customized visual representations with related interaction options (Zimmermann, 2017). The design of visualization should serve to convey complex information to academic users as well as interested laypersons. The virtual map forum finished this task well. The portal offers several controls (see fig. 3.7). A time slider can restrict the time period in which the maps were drawn. The filters on the left allow the users to choose the types of map and the search function allows the possibility to focus on specific locations and map types. The essential function of the portal allowed the author to obtain the selected map sheets by WMS or WCS interface or by downloading the TIFF file.

3.4 Pre-processing

Firstly, the colour of maps is homogenized. As the maps cover the wide state range and, as described, are with large scale and incredible details, coupled with labour and technical limitations at the period, the Berlin Copy was completed from 1780 to 1806 for more than 20 years. The craft work with such a time span brings non-standardization. Although the cartographers tried their best to ensure the aesthetics and readability of the maps, the differences in colour and shape between the text and the maps will undoubtedly bring a lot of challenges for generalization learning. Figure 3.8 offered examples of two maps drawn in different time slots, where the background colour and font variety are apparent. To homogenize the hue and increase the contrast, the Historic

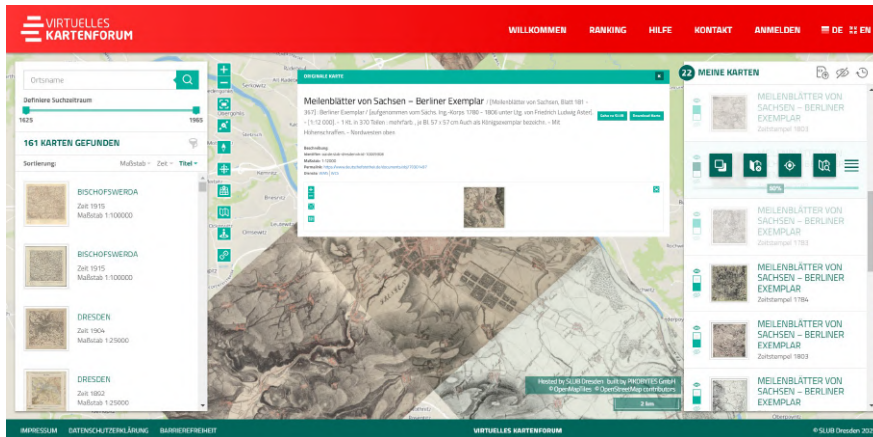


Figure 3.7: User interface of virtual map forum 2.0



Figure 3.8: Parts of two neighbored map sheets from Saxon Mile Sheets. The left one was mapped in 1803, and the right side is one from 1784. The place name "Groß Naundorf" was annotated on both parts but in different sizes and styles.

Map Semantic group developed a script to automatically set the black area and white area to stretch the colour histogram value.

Secondly, the maps are re-projected to an orthographic view. The maps retrieved from the Virtual Map Forum are oblique images, which means distortion in features. So, the colour-homogenized map scans are converted to the conformal projection system and re-sampled to 1 meter.

Figure 3.9 shows a map sheet downloaded from the Virtual Map Forum and re-projected and colour homogenized sheet ready for further processing.

Thirdly, the maps are selected and cropped into smaller patches to increase computational efficiency. The Historic Map Semantic team provided software for obtaining a collection of patches from map sheets. To ensure the model can learn various features sufficiently, the patches selection was performed automatically, accounting for the full range of hues and densities found in one map sheet. Map samples composed of patches of 250 by 250 pixels each can be created from the georeferenced map files

3 Material

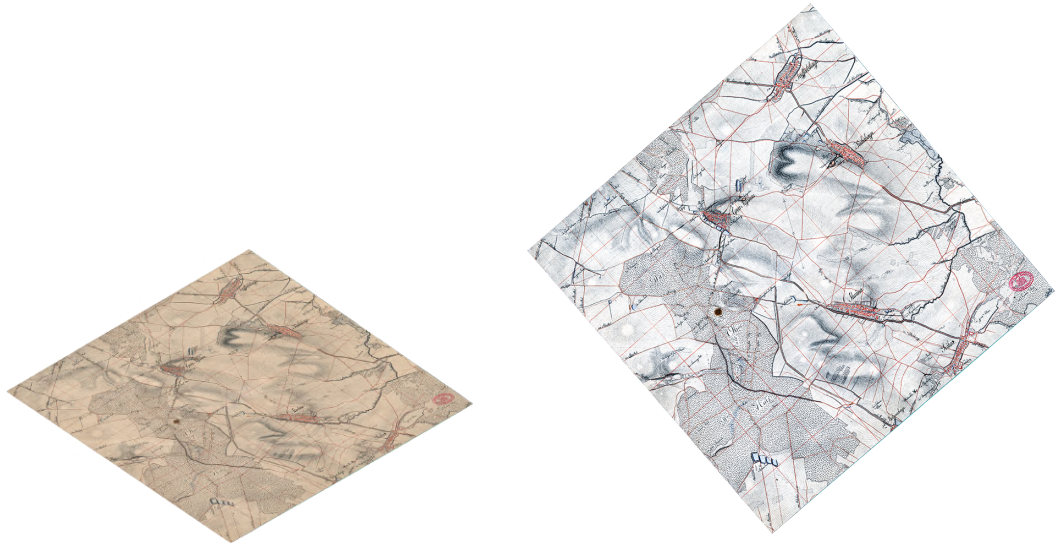


Figure 3.9: An example of an original map sheet downloaded from the map forum (left) and after re-projection and colour homogenization (right).

with a pixel dimension of 1m by 1m. For further use as training and testing data, the individual patches can be regrouped from several map sheets according to demands.

4 Methodology

The author first tested the performance of the existing pre-trained text detection models introduced in section 2.2.2. The author would introduce the main processing in section 4.1 and more details about the methods adopted in the following sections. The author has documented the processing pipeline in a separate file as an instruction for further usage.

4.1 Workflow

The related work in OCR domains and previous studies of deep learning-based methods for text extraction from historical maps offered a solid foundation for this thesis. Based on all the reviewed frameworks, this thesis proposed a procedure for unlocking place names from Saxon Mile Sheet (see fig. 4.1).

The processing was divided into three parts in general (see fig. 4.1). From data preparation (RQ1.2) to text detection (RQ1.3), then, at last, the recognition (RQ2.2), the outcome of each part was taken as the input for the next. Since sequential execution brings the potential problem that the downstream stages highly depend on the results of upstream implementation, there are evaluation and calibration steps in each part. Steps were revisited when the outputs were not satisfied.

4.2 Data Preparation

The processing map sheets were selected from different parts of Saxon to get sufficient diversity. After reprojection and colour homogenization (see section 3.4), nine map sheets were grouped into the training set, and four map sheets were used for testing. The training maps were further cut into patches. The author ensured the training set and testing set were independent.

4.2.1 Training Dataset

The 1.2 research question of the thesis was about whether synthetic data can help improve the efficiency of data preparation while preserving the high performance of the deep learning model. In order to answer this, the models were trained on both simulated data and manually labelled actual map patches (see fig.4.2).

The ground truth data was from the selected training patches with annotation. The toponym in the patches was manually pixel-wise extracted and binarized as foreground

4 Methodology

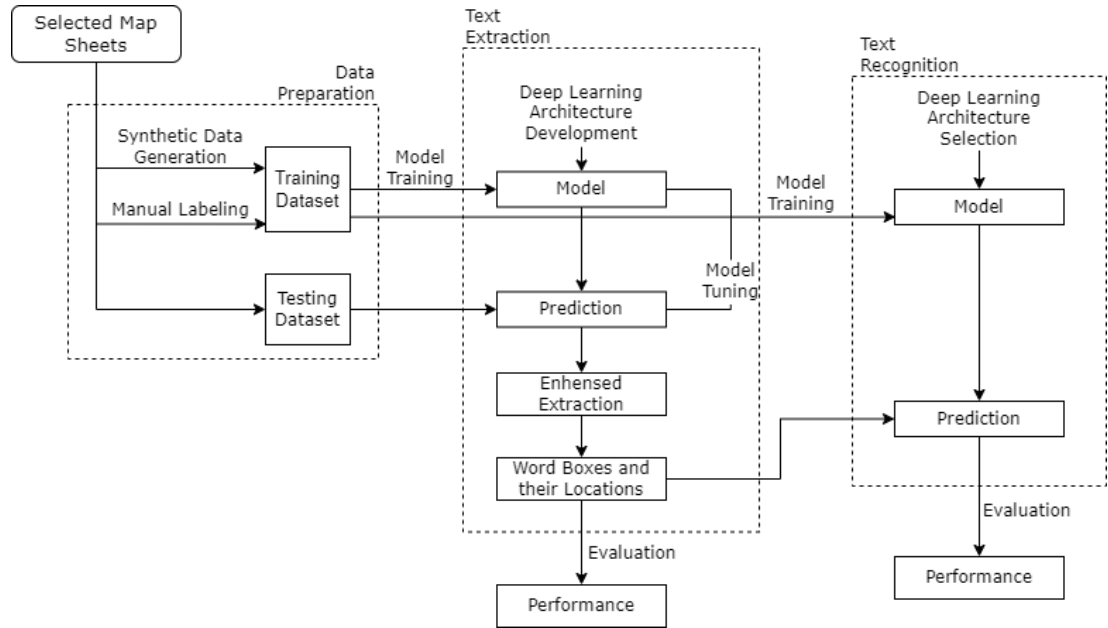


Figure 4.1: General Processing Workflow

(white) in image editing software (see 4.2a). The manual labelling was time-consuming and can be error-prone in extracting the precise shape of the text.

The thesis applied the most straightforward synthesis to the data (see fig. 4.2b), merging a text layer and a background image layer. To begin, the author filtered out neutral background images covered with various feature types (see fig. 4.3). Those are patches without any text information. Extremely dark backgrounds, for example, covered with very dense slope shades, were excluded from generations of true positive patches but were sorted into true negative training. This is because the place names on these kinds of backgrounds can not even be recognized with human eyes. Also, the author decided to focus on the large and middle-sized toponyms (see section 3.3), which mostly only intersect with simple features. The text is randomly chosen from a list of possible place names or a random sequence of letters, but ensure all letters in German are traversal. The text was rendered with a customized antique font and at font size 80. The rendering colour is randomly from dark brown to black, and a small part of light brown. The vector graphic was then rasterized. As presented in section 3.4, the largest and middle-sized toponyms are placed in roughly one direction. Therefore, the text was rotated 39 degrees counterclockwise before being overlaid on the background layer using Python Imaging Library. As antialiasing was not included in the font, and the author intended to have binary images as training input, a morphological closing operator with a kernel of 1*2 was introduced to smooth the jagged edge.

True-negative patches were selected from the list of neutral images of each category (see fig.4.3). The initial selection was even. After the first training was conducted, based on the training results, if there are features with particularly problematic predictions, for example, if a large number of tree elements are misclassified as text, then the training data would increase the number of patches containing the feature.

Different augmentation strategies were applied respectively to true-positive and true-negative data. For true-positive images, the author wanted to preserve the orientation while all other non-textual features can be in any direction. So, the common augmenters were adding random blurring effects, resizing and translating and random cropping. For true-negative data, additional flipping, rotation and contrast modification were executed. The operations were conducted with the help of the python library imaging.

In the implementation stage of model learning, the input data were further extracted from augmented samples to match the size limit of the architecture. Part of extracted

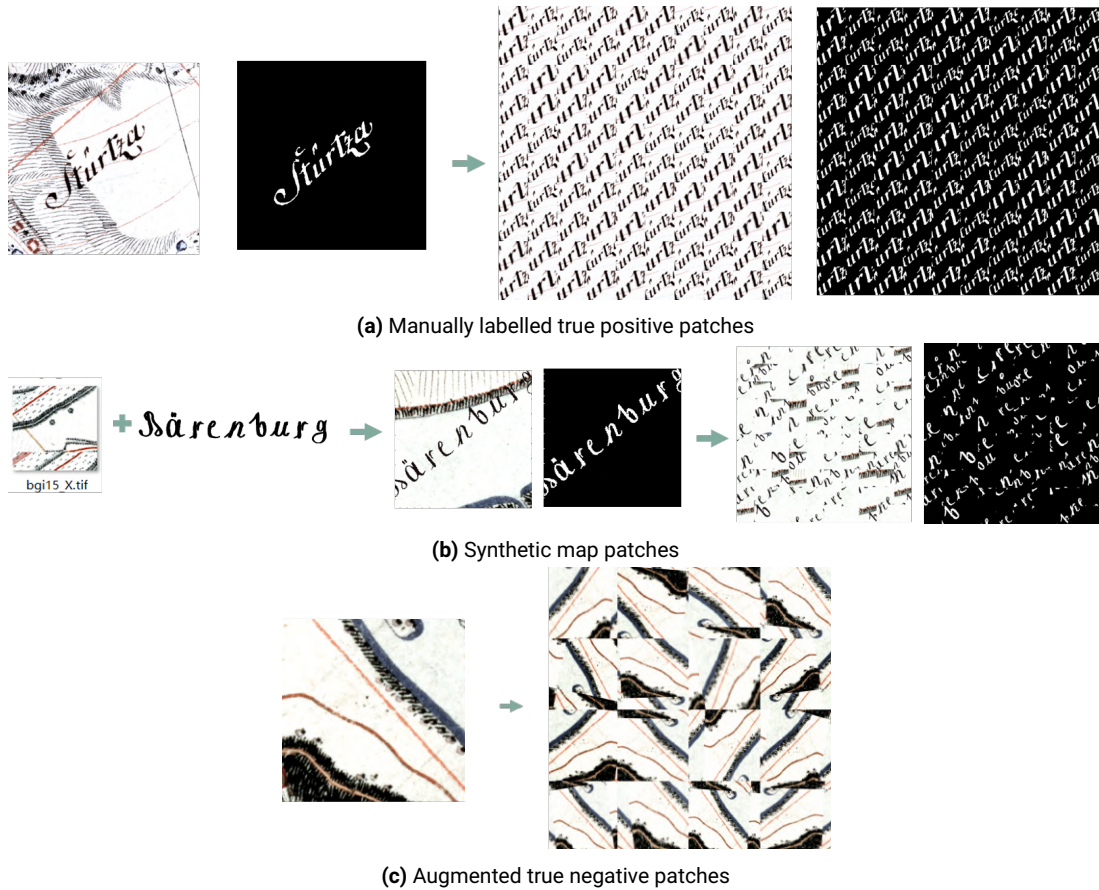


Figure 4.2: Illustration of the data preparation procedure. The size of the images is not proportionally scaled.

4 Methodology

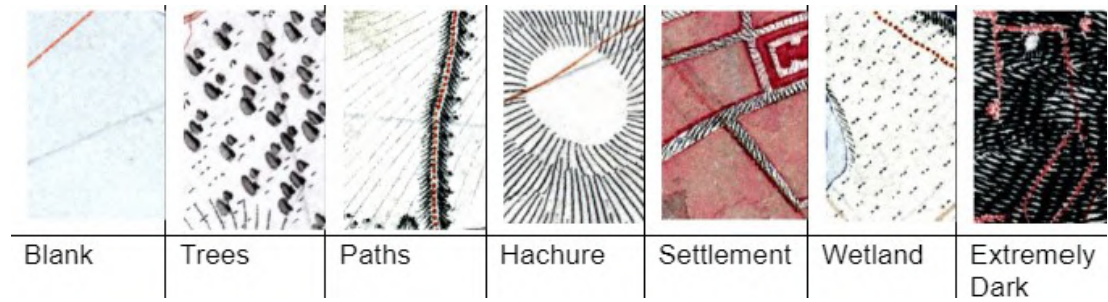


Figure 4.3: Patches extracted from maps were classified into different feature groups (see section 3.4)

samples was saved separately and used as the validation set.

4.2.2 Testing Dataset

After reprojection and colour homogenization (see section 3.4), a piece of the Saxon Mile Sheet is around 9600 to 10000 pixels in width and height. Due to the computational limitation and to increase the efficiency of model tuning, the trained model was first tested on a collection of clipped map sheets. The author cropped regions of text with different features on the background from the selected testing maps and placed them on a white background canvas. The actual map sheets were tested after optimal models were selected. Figure 4.4 shows the image used to test the models.

4.3 Toponym Extraction

The overall processing steps of text extraction are as shown in figure 4.1. The processing started by separating text from the graphical background using the principle of binary semantic segmentation. The machine learned the features of text elements from input augmented synthetic or manually labelled training patches. The map images were transformed into black-and-white text masks. Then morphological operators helped to merge the extracted dispersed characters to text regions and then from which to draw the contour line. The image was then cropped to word boxes containing the detected toponym.

4.3.1 Trained Models from Existing Framework

For testing text detection on off-the-shelf frameworks, models were selected based on three factors. First is the availability of the model and code. The second was the robustness, whether it has been tested on different datasets in general and has overall good performance. Thirdly, the model's computational cost was also considered to



Figure 4.4: A set of cropped text areas use for model testing.

avoid memory error due to hardware limitations. Therefore, the author was not able to compare all the algorithms reviewed in section 2.2.2.

MMOCR (Kuang et al., 2021) is a powerful open source toolkit for text detection, recognition and understanding. The toolbox is a part of the OpenMMLab project. Compared with other open source OCR projects (for example, PaddleOCR, and EasyOCR), the available models are the most and the latest. Fourteen state-of-the-art algorithms are integrated within a unified framework. The paper takes full advantage of the pre-trained models built in the toolbox to test how well they can be tailored to complete the text detection tasks on Saxon Mile Sheets.

The model for comparison is firstly the segmentation-based DBNet++ trained on the ICDAR2015 dataset. ICDAR are a series of popular benchmark datasets published through the International Conference on Document Analysis and Recognition (ICDAR) organization. ICDAR2015 is one of it which contains incidental scene text data labelled with quadrilateral text boxes in multi-orientation. Another segmentation-based model tested was TestSnack trained on the CTW1500 dataset, which is a curved text database. The third model chosen was the regression-based method DRRG also trained on the CTW1500. The hybrid method, the FCE model, trained on ICDAR 2015, was also tested.

4 Methodology

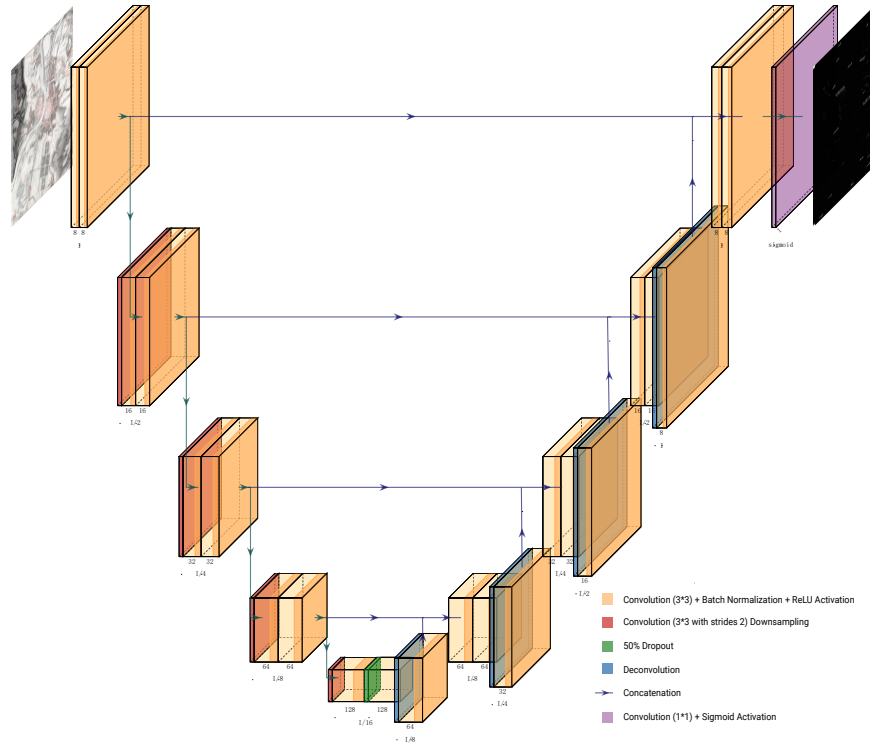


Figure 4.5: The architecture of implemented U-Net

4.3.2 U-Net Architecture

The deep learning structure adopted in this thesis is the U-Net architecture (Ronneberger, Fischer, & Brox, 2015). U-Net is one of the early algorithms developed based on fully convolutional networks for semantic segmentation. The "U" in the name comes from its symmetric U-shaped structure as visualized in figure 4.5. The right side of the model is described as the encoder, the so-called contracting path in some studies, and the left side is the decoder, the so-called expansive path. A crucial component connecting each level of the encoder and decoder is the operation of copy and concatenation. This aspect makes it a prominent difference to the fully convolutional network.

The encoder downsamples the feature maps, which means reducing spatial data and extending feature data. The compressed path consists of four level blocks, each of which uses two effective convolutions and one downsampling layer, and the number of feature maps is multiplied by two after each downsampling, resulting in the feature map size variation shown in the figure.

In order to gain location information, the decoder reconstructs the image to return the segmentation map with the exact dimensions of the original image. The blocks

	Time of ten runs
CPU	0.7597459
GPU	0.0271611
GPU speedup over CPU: 27x	

Table 4.1: Time (s) to convolve 32x7x7x3 filter over random 100x100x100x3 images (batch x height x width x channel).

of the expansive path start from the size expanding with the deconvolution layer. The concatenation connection crops and carries the feature map from the corresponding downsampling step to the decoder. In the block of the decoder, the two compressed feature maps are merged together.

Each level block contains additionally batch normalization to reduce the internal covariant shift and accelerate the training. A 50% dropout probability was added at the deepest level to reduce overfitting.

UNet is chosen primarily for its ability to learn from a small training set. The feature map acquired from the deeper network layer has a larger receptive field, so the shallow layers are sensitive to texture and the deep layers concerned with the shared features, and UNet is able to learn from both. UNet is effective with its lightweight and simple structure.

The thesis uses a U-net architecture computed with the Keras library, and the training was implemented on the GPU. The speed of the GPU and CPU processor was compared on the local laptop shown in table 4.1. The standard max-pooling layers for downsampling were replaced by convolutional layers with a stride of two, which further extended the ability of feature learning. Same in upsampling, transpose convolutional layers were executed.

The input image size for the network training was 128 * 128. The number of channels of the first convolution was encoded to 8. Then following the sequentially downsampling, with each convolutional block, the number of kernels doubled and went to 128 at the deepest level. Compared to the 1024 feature maps in the original paper, the model was simplified due to a trade-off between the complexity of the model and the size of the input image. As the purpose of the network was to separate the text feature from non-text, the final output is only one feature map.

4 Methodology

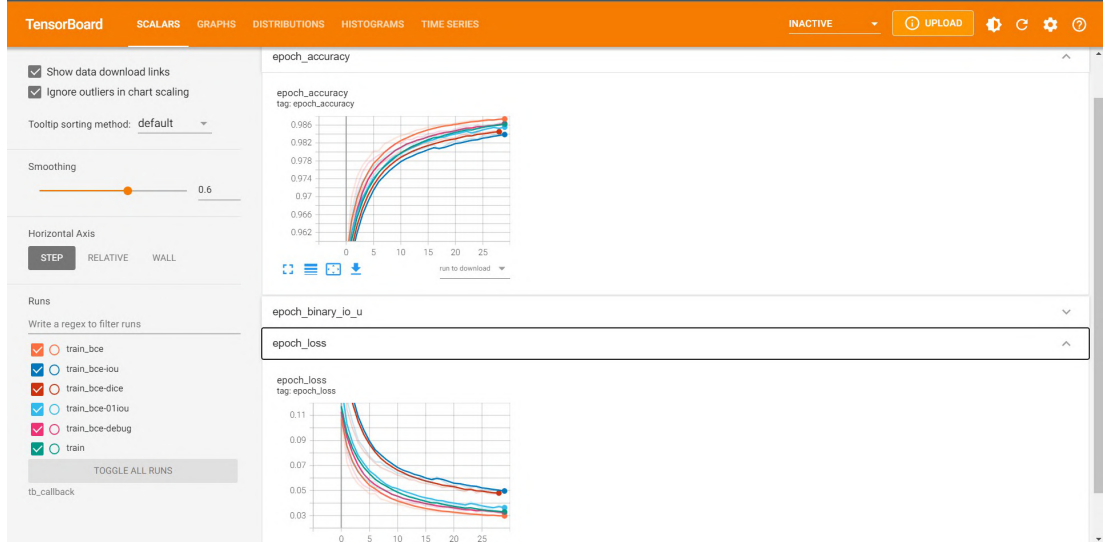


Figure 4.6: Model performance tracked and visualised on TensorBoard

4.3.3 Performance Tuning

Deep learning is complex, and even the relatively simple structure of the Unet architecture has many hyperparameters. Choosing the already appropriate hyperparameters at the beginning is undoubtedly impossible, so after the network architecture is built, the researchers always have to tune the hyperparameters to improve the performance.

The tuning process of deep learning is more empirical than theoretical, which in other words, is based on experts' experience and can only achieve by repeating the iteration until the optimal results are found. Therefore it's important to track the performance and have a direct comparison of the change in accuracy. Tensorboard is a visualization toolkit that can host and track the training process of deep learning models. There are diverse functions embedded in the platform, while for this thesis, the writer mainly used the scalars dashboard, which visualizes the loss and metric of each run (see fig.4.6).

Table 4.2 records the training schedule of the network in this thesis. The author tried to improve the model performance from three perspectives. Firstly, the author tune parts of the hyperparameters; secondly, the amount and quality of training data were explored; and thirdly, different loss functions were tested.

The most tuned model hyperparameters are learning rate and batch size.

The learning rate controls the magnitude of network gradient updates in training. The optimal learning rate should accelerate the training of the model and obtain satisfactory accuracy. If the learning rate is too large or too small, it will directly lead to the divergence

4.3 Toponym Extraction

Experiment	1	2	3
Learning rate	1.00E-03	2.00E-03	1.00E-02
Batch size	5	5	5
Loss function	Binary Cross Entropy (BCE)	BCE	BCE
Input data size	(128, 128, 3)	(128, 128, 3)	(128, 128, 3)
Number of training patches (true positive + true negative)	15 + 15	15 + 23	15 + 25

Experiment	4	5	6
Learning rate	5.00E-03	1.00E-03	1.00E-03
Batch size	5	3	5
Loss function	BCE	BCE	Intersection over Union (IoU)
Input data size	(128, 128, 3)	(128, 128, 3)	(128, 128, 3)
Number of training patches (true positive + true negative)	25 + 25	35 + 37	49 + 37

Experiment	7	8	9	10
Learning rate	1.00E-03	1.00E-03	1.00E-03	1.00E-03
Batch size	5	5	5	5
Loss function	Dice Loss	BCE + 0.2 * IoU	BCE + 0.2 * Dice	BCE + 0.2 * IoU
Input data size	(128, 128, 3)	(128, 128, 3)	(128, 128, 3)	(128, 128, 3)
Number of training patches (true positive + true negative)	49 + 37	49 + 37	49 + 37	49 + 37

Table 4.2: The training schedule of the recognition network

of the model. With the Adam optimizer, most projects tested the initial learning rate in the range [1e-3, 1e-2]. The author tested the learning rate of 0.001, 0.002 and 0.01. The optimal results were performed with 0.001.

The batch size determines the number of samples processed in one iteration. For small data sets, full batch learning can be conducted, but for larger data, loading all the data at the same time will cause memory pressure. The appropriate batch size can improve memory utilization and speed up processing. The author tried to reduce the batch size to prevent memory error, but the model was not able to converge inside the limit epochs with a smaller batch size. The optimal batch size was tested to be 5.

The primary strategy of machine learning is to minimize the loss function, to have the model reach a state of convergence and to reduce the error in the model's predicted values. Therefore, configuring the loss function is a crucial step to ensure that the model works in an expected way. Different loss functions can have a significant impact on the model.

The default loss function for binary classification questions is cross-entropy. Cross entropy can be briefly summarized as a calculation of the difference between two

4 Methodology

probability distributions. It was built upon entropy, which is a measurement of uncertainty in information theory. The Binary Cross Entropy Loss function is:

$$-(y \log(p) + (1 - y) \log(1 - p)) \quad (4.1)$$

where y is the true value, can be 0 or 1, and p stands for predicted probability for the class.

Dice loss is a function used to assess the similarity measure between two samples, taking values in the range of 0 to 1. The larger the value, the higher the similarity of the two values and its basic definition for binary classification is as follows

$$DICE = 2 * \frac{|A \cap B|}{A + B} \quad (4.2)$$

Jaccard similarity index, also known as Intersection over Union (IoU), is widely used as an evaluation metric, especially for the object detection task. It can also be computed as a loss function for binary semantic segmentation when the element to be extracted is much smaller than the background elements. The IoU score and loss are calculated respectively as followed:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (4.3)$$

$$Loss_{IoU} = 1 - IoU \quad (4.4)$$

where A and B are the ground truth set and prediction set.

The line chart in figure 4.7 compares the performance of the last three runs trained with different loss functions. The models trained with all three loss functions show a very high accuracy after iteration of several epochs. This is because, pixel-wise, compared to the large non-text region, the text elements only occupy a tiny ratio. Therefore, the accuracy can easily reach a high level statistically, even though many text features are not extracted. That's why the IoU score should be introduced to assess the model.

While the model trained with BCE shows the highest accuracy and lowest loss statistically, the prediction of models trained with BCE and IoU showed strength in precise shape extraction (see section 5.1.2). Therefore, the idea would be to combine different models to get a prediction that outperforms all single results. Sarker (2021) took model ensembling as a potential direction of deep learning research. A group of neural networks can be trained with different parameters or independent subsampling training datasets. A promising solution for model uncertainty can be hybridization or ensembles of such models. By allocating weights based on each model's performance,

the author attempted to optimize the predictions made by an ensemble of models from several trained models. Selected predictions were calculated to weighted average as a final extraction output.

The author also tried to train the network with 512*512 sized image input. As for 128*128 patches, some of the words were cut into strokes or points, which might cause different feature maps. However, due to the computational capacity of local laptops, memory errors interrupted the execution.

4.3.4 Post-processing

After segmentation results were acquired, the author intended to enhance the image as a post-processing step of text extraction or from the text recognition side as a pre-processing step to prepare the input for OCR tools. Morphology is a basic technique for image enhancement. The primary operations are dilation and erosion. Having different process sequences of these two basic operators, morphological opening and closing operations perform with different effects. Opening operators conduct first erosion, then dilation. It functions to break the connections and eliminate small isolated pixels. In this thesis, it was applied to remove the non-textual noise. The closing can smooth the contour and fuse the narrow breaks. For the characters, it helps to fill the holes inside the text shapes. So, the author applied to open followed by closing to get a cleaner shape of the extracted letters. Figure 4.8 shows an example of an extracted text element and the image after enhancement.

4.3.5 Word Box Localization

The image masks were then processed to get the word box. For easier detection, the image was first rotated to turn most of the text in the horizontal direction. Morphological closing with horizontal kernel operated to fuse the disbanded text letters to a single word

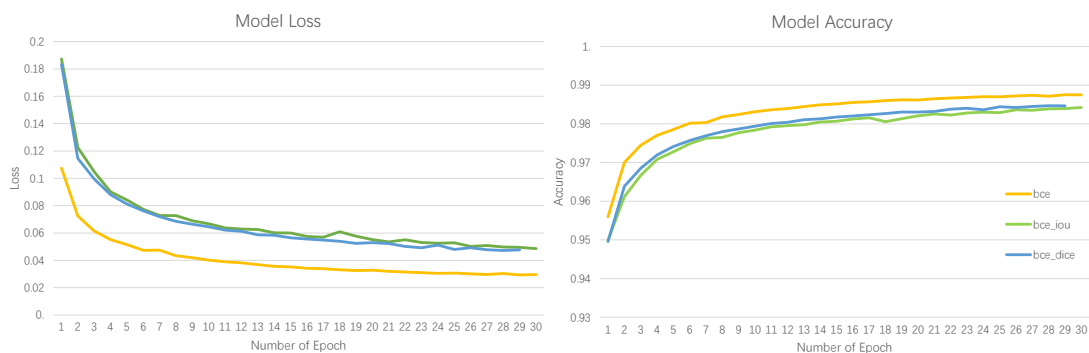


Figure 4.7: Comparison of model performance with different loss functions.



Figure 4.8: Text extracted and after enhancement

region. Then the rotating calipers algorithm was implemented with the OpenCV library to find the smallest bounding box of the word regions. In order to make the toponym more completely extracted, the authors have performed a slight scaling operation on the enclosing rectangle.

The outputs of this step were:

1. Visualization of detected word boxes on the original image. Text boxes were marked on the original coloured map to help the viewer see the direct visual results of the text detection.
2. A greyscale and an enhanced binary version of the text image were ready for text recognition. The greyscale version contains more textual information, while the enhanced binary version offers cleaner shapes. The selection of version using as input data would depend on the requirement of the OCR tools.
3. Cropped slices were prepared for the OCR tool (MMOCR), which takes cropped text tiles as input.
4. The location information of the text was also recorded. The authors extracted the geographic coordinates of the text box's central points and queried the address through the open source geocoding API, Nominatim. This will prepare for possible contextual analysis for text recognition correction or potential study of toponym change in linguistics in the future.

4.3.6 Evaluation Metrics

To assess the performance text detection model, we employed three metrics. They are precision, recall, F-Score and Intersection over Union. In the literature, these metrics are frequently employed for object detection topics. The author was not interested in pixel-level misclassification of the text elements but rather concerned with the word as

an entity. So all the metrics were calculated at the object level based on the detected word boxes.

The intersection over Union (IoU) metric was computed to evaluate how complete the place names were extracted with bounding boxes. The ground truth is the actual text regions marked manually. The prediction area is the area within the minimal bounding boxes. The score was calculated by dividing the intersection of these two regions into the union of these regions (see equation 4.3).

Recall, Precision and F-Score were calculated as follows:

$$Precision = \frac{TP}{TP + FP} \quad (4.5)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.6)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \quad (4.7)$$

4.4 Toponym Recognition

4.4.1 Training Datasets

For text recognition, the authors also attempted to use synthetic data for training. Text of random font size was generated on the page using the customized antique font of map series and rendered to graphic, and the random text was saved directly as labels.

The other set of training data came from one part of the manually labelled map patches from the data preparation step. Another part was selected and annotated from the prediction result of text detection.

The datasets created by the author were in small sample sizes. Although the samples of synthetic data can be theoretically infinitely, training data lacked sufficient variation. Therefore the author used pre-trained models from Transkribus' database as a supplement. This model trained on German historical documents from the National Library of Australia claimed to have good accuracy in recognizing German fraktur from the 19th to the 20th century.

4.4.2 Deep Learning-based Text Recognition Tools

Although the existing deep learning-based OCR usually contains the function of text detection already, they usually can only process the text on simple backgrounds. The testing on trained detection models also proved that the end-to-end OCR systems are not sufficient to finish the text extraction task on Saxon Mile Sheets. Therefore, the pre-processing of maps to separate text and graphics and to remove the majority of distraction noise is a necessary step before starting the recognition.

The recognition was conducted in MMOCR and Transkribus (Kahle, Colutto, Hackl, & Muhlberger, 2017). The recognition was conducted in MMOCR and Transkribus. Transkribus is a comprehensive platform for text recognition, having a particular focus on historical documents. It is open source, has detailed user instructions, and is, therefore, easy to use. Compared to MMOCR, the advantage of it is the multiple pre-trained German transcription models publicly available.

The author used the HTR+ engine embedded in the Transkribus tool for recognition. HTR+ is an improved version of the handwritten text recognition engine from the CITlab team (see section 2.2.2). The author first train on the image with a small amount of training data and on the base of the pre-trained model. As the error rate was high, proving that the pre-trained model was not suitable for the task, the author started to increase the training data input and trained the model from the beginning.

In mmocr, the classical CRNN+STN model was chosen for text recognition. A trained ABiNet model was also tested to see the alternative option other than CRNN. Both models are trained on the Syn90k dataset. This Syn90k (MJSynth) dataset contains 9 million images consisting of 90k English words.

4.4.3 Evaluation Metrics

For assessing text recognition systems, the character error rate (CER) and word error rate (WER) were calculated. They measures how much text in the handwriting was incorrectly interpreted by the HTR model.

$$F1 = \frac{errors}{correct + errors} \quad (4.8)$$

It is controversial to use WER for model evaluation. Because regardless of the length of the word, if one letter of it is incorrectly interpreted, the entire word is marked as a wrong recognition result. Even if the model has a low CER, its word error rate may be high. Moreover, WER does not give comparative results for individual characters and is therefore used relatively seldomly to evaluate recognition models. However, WER

can imply whether the recognized text is possible to be successfully searched or fuzzy matched in a database such as a gazetteer, so this thesis still calculated the WER.

Additionally, the author visualised the confusion metric for recognition of each letter (case-insensitive) to analyse the error distribution. Confusion matrices are useful for determining pairwise which letters are better distinguished and which of those cause the most confusion.

5 Results and Discussions

In this chapter, the authors presented the results of the workflow implementation with graphs and discussed the results. The authors shown section by section, how well the training of synthetic data performed, how well can the trained off-the-shelf models and the proposed UNET model extract the place names, and how is the performance of selected recognition models.

5.1 Text Extraction

5.1.1 Pre-trained Models

Figure 5.1 shows the detection results of the four selected models on the test images. The detected text region is framed with green quadrilateral or irregular box lines. Due to page limitations, the images are scaled to show only the more problematic areas of each model prediction.

Text recognition does not distinguish size on the generic model because the general models are not specifically trained. The smaller text should theoretically be recognized as well.

Overall, the performance of the four models varies and all have specific types of errors.

In the DBNet++ model (see fig.5.1a), black line elements such as some river banks, slope lines, and red geometric elements of residential areas are incorrectly identified as text. For the toponym in larger fonts and spacing, there are one-to-many errors, which means the word entity is truncated into multiple text boxes. The bounding boxes do not frame the text completely. Some characters of the place names are located outside the detected bounding boxes. The false negative error happens on one test crop, and all the other place names are at least partially detected.

TextSnake has a good performance on toponyms when the characters have similar height. The minimal enclosing box are precise around the text region. For large text, not all the character are correctly located. Type 2 errors are found on densely distributed point features.

The DRRG model has a many-to-one error on the detection of smaller text, where different place names are framed into the same text box. But it performs well on text area detection of medium and large-sized place names. The text shapes in the box are relatively the most complete. However, there are three slices on which the text is ignored. No background elements were classified as text.

5.1 Text Extraction



Figure 5.1: Detection Results using Pre-trained Models. Only problematic regions of test images are shown due to space limitation.

5 Results and Discussions

The detection of the FCE model is more problematic in comparison to the three. The network is sensitive to smaller and densely distributed characters. The largest place names are not detected or are just partially framed. In particular, the taller letters such as "h" and "f" are not enclosed in the bounding contours. Incorrectly classifications are found in slash stripe features and polygon features of settlements.

In comparison, DRRG performs best in separating text and non-text features. The two segmentation-based models, DBPP and TextSnake, are able to correctly find more text regions, but both interfered from other elements, and type II errors appear more often.

There is text not located in all of the four models. It can be seen that the direct use of trained models for toponym detection on historical maps is not satisfactory. The results are not sufficient to be the input into OCR tools for recognition. Thus, this motivates the author to train a specific model for the Saxon Mile Sheets.

5.1.2 U-NET Pipeline

The U-Net training pipeline includes the stages of training a model, model tuning, choosing the models to predict, ensembling predictions and extracting the word boxes.

Figure 5.2 to figure 5.5 present the model tuning process and the predicted results. Results were selected from one run of the total 30 epochs.

Figure 5.2 shows the text shapes are more precisely extracted using optimized training data. A series of optimization applied to training data, including the adjustment of augmentation algorithms, choice of binarization threshold, and a more diverse selection of background features. This indicated the crucial role of synthetic data generation approach in having a good segmentation result.



Figure 5.2: Predictions before and after improvement of training data

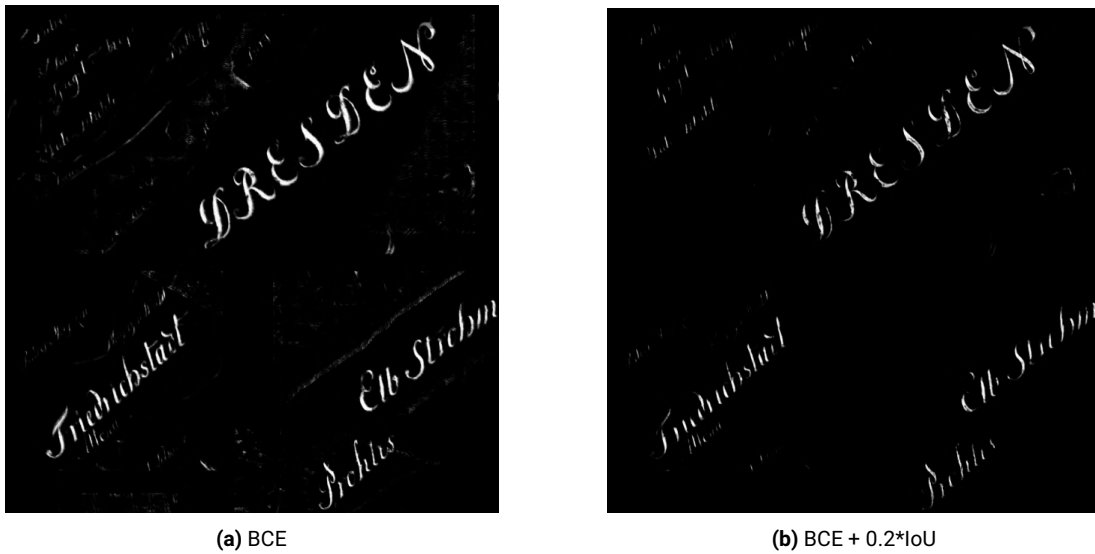
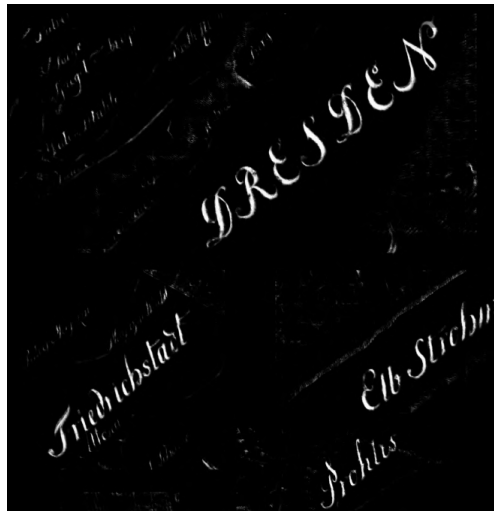


Figure 5.3: Predictions from models trained with different loss functions (BCE and BCE+0.2*IoU).

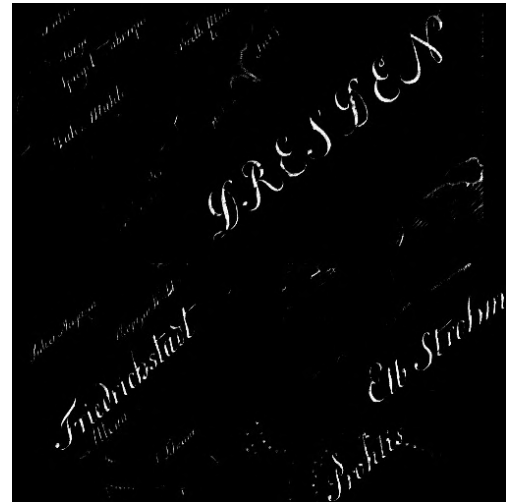
A visual comparison of models trained with different loss functions can be found in figure 5.3. BCE preserves the text shapes better, whereas the combination with IoU brings a better background removal. The elements that are misclassified to the foreground are mainly dark dot and line features that have similar colour and shape to the text. The contradiction always exists that the more accurate the shape extracted, the more noise is also classified as the text element. The same situation can be seen in the trained models (see fig.5.1). On one side, the author has to find a balance between shape extraction and background removal. On the other side, model ensembling can be a solution to combine the strength of two models. In the thesis, predictions from both BCE models and BCE+IoU models are kept and hybridized for post-processing.

In order to provide an impression of how synthetic generation can help to save the effort in deep learning, the control group is the model trained on small amount of manually labelled ground truth samples. Visually there is no significant improvement in balancing background removal and shape extraction (see fig.5.4). The text strokes are still defective in the segmented image. This also affirms the possibility of replacing manual annotation with synthetic images. The automatically generated training data is at least comparable to the small-scale manually annotated data. They do not differ much in prediction performance, so the synthetic data can significantly reduce the labour cost compared to the difficulty of data acquisition.

The defect shapes in prediction are usually faded coloured strokes on original map. An example is the letter "w" and "z" in the toponym "Lockwitz", as framed out in figure 5.5. They can be an indicator of how well the model is performing in preserving the text



(a) Synthetic training set



(b) Manually labelled ground truth

Figure 5.4: Predictions from models trained with synthetic data automatically generated and trained with manually labelled true map patches.

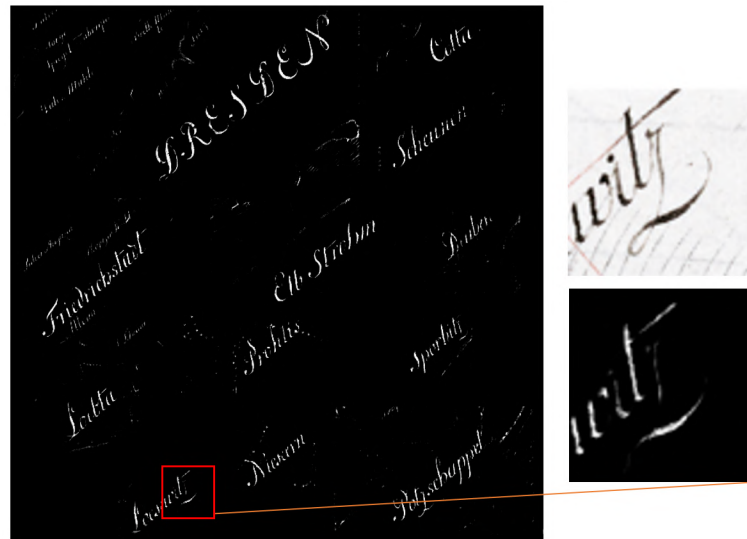


Figure 5.5: Prediction trained with manually labelled data

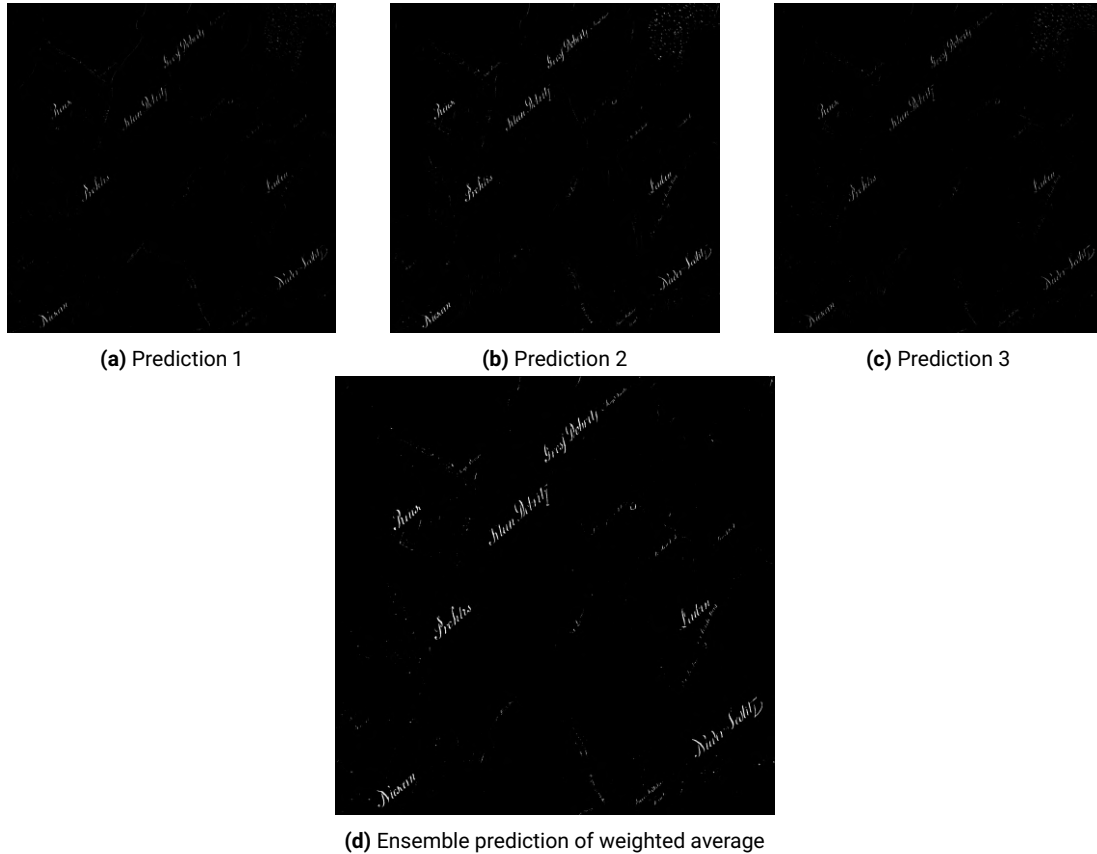


Figure 5.6: Predictions from models trained with synthetic data automatically generated and trained with manually labelled true map patches.

shape. The chosen model is expected to extract at least a fuzzy contour of stroke as "z" in order to make it possible to be read and interpreted by the recognition.

Based on the results on the test image, the authors chose five BCE and BCE+IoU trained models to conduct prediction on the actual map sheets. The figure 5.6 (a)-(c) depicts parts of the results of a map sheet. The displayed sheet is mapping the region around Niedersiedlitz. The three predictions are the optimal ones among all the outputs, and they are respectively strong in shape extraction or in text/graphic separation. The author wanted to strengthen the text shape while keep the noise excluded from the foreground. Therefore, the weights of 0.3, 0.5, and 0.2 were assigned to the prediction according to their ability of background removal and then the mean values were calculated. Figure 5.6d shows the ensemble result. In the stacked result, the elements with a higher probability of being predicted as text were further highlighted and had a stronger contrast with the noise. This allows easier differentiation in the subsequent post-processing steps.



Figure 5.7: Detection output for map "Neustadt".

Map	Precision (%)	Recall(%)	F1 (%)	IoU (%)
TestCrop	46.15	100	63.16	71.34
Neustadt	94.44	100	97.14	63.88
Woelckau	45.45	100	62.5	44.97
Nidersedlitz	62	100	76.54	57.19
Fuchshain	47.06	100	64	1.14

Table 5.1: The evaluation of text box detection on different map sheets. Map names are the major cities in the mapping area of the pieces.

The ensemble results are processed to acquire the text regions. Defective text strokes are further restored in post-processing. Both the enhanced bicolour shapes and the original RGB image were cropped to separate image files. The figure 5.7 shows the output of one of the maps sheet. The text box nicely frames out the detected toponyms. Only the extraction of a few small place names is incomplete, which is also unexpected as the model was trained only with larger font size. Only the middle part of those tiny street names is intercepted in the text boxes. The background dots and lines are removed. The extracted text shapes have good readability, but some of the "e" and "t" letters are incomplete, and might cause confusion in the recognition.

5.1.3 Evaluation

Table 5.1 shows the evaluation of the prediction results for all tested maps. The performance of the models varies on different maps. However, all recall scores are 100%, indicating that all the true positives are correctly detected as positives. The mapping region of the sheet can be one of the factors that caused the gap. Map sheets "Woelckau" and "Fuchshain" are spatially distant from the other maps and show a lower evaluation score. The cropped collection in the testing image did not cover sufficient diversity. This suggests that more various training data inputs and more extensive testing on maps from wider regions are necessary. But even the lowest F1 score is much higher than the performance of CNNs tested on Austro-Hungarian historical maps (Can & Kabadayi, 2021).

The IoU result computed is inaccurate for different reasons. The bounding boxes manually drawn might not be the minimal enclosing contour. The manual marking is mostly horizontal to the visual text central line, while the minimal bounding box computed could be rotated. The angular difference then brings the misaligned areas. Secondly, manual labelling only focuses on the large and middle-sized toponyms, whereas the deep learning model also extracts the small ones. The large percentage of extra extractions marked as false positive caused the low value in IoU and precision. Especially for the map sheet Fuchshain, where smaller annotations for routes and



Figure 5.8: Detected small toponyms on map "Fuchschain"

landscape are written in the dark solid font. Even after filtering out the small text by setting a threshold of the text area, the network located many small toponyms and extracted the shape clearly (see fig 5.8). Therefore, the prediction for the map sheet "Fuchshain" with extraordinarily low IoU cannot be claimed as a complete failure. Almost all text boxes contain textual information. The mixture of toponyms in different font sizes indicates that the proposed network cannot differentiate text with size variance in the thesis implementation.

One adaptation for IoU in text detection is that use the over lap ration to measure the true positive and false positive (Khan et al., 2021). True positive is considered when the value of IoU ratio is greater than the selected threshold. This method is adopted in the evaluation of ICDAR 2015 dataset. This might be also a good metric to assess the model proposed.

5.2 Text Recognition

5.2.1 Models Performance

The extracted text images were then passed to the recognizers. Table 5.2 compares the performance of selected recognition models. The implementation of CRNN+STN

Model	Input	CER (%)	WER (%)
CRNN+STN	RGB	16.24	76.92
	Extracted Binary	17.95	76.92
ABiNet	RGB	36.75	92.31
	Extracted Binary	41.03	92.31
HTR+	Extracted Binary (whole page)	88.89 -> 70.09 -> 52.51	100 -> 100 -> 95.65

Table 5.2: The performance of trained models on reconizing the toponym. The arrow in HTR+ column indicate the change of training dataset.

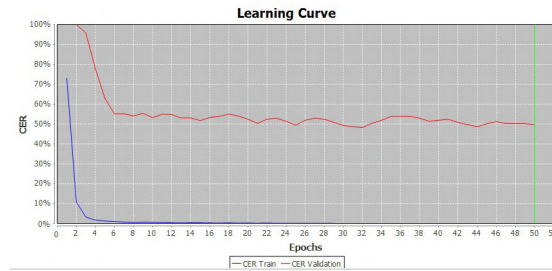


Figure 5.9: Learning curve of the recognition model trained with synthetic data and validated with toponym patches. The character error rate decrease in runs but the gap between training and validation set is so large.

and ABiNet model with MMOCR took cropped images as input. Transkribus software could not recognize the text inside cropped text boxes, so we used the whole binary image of the segmented result as input. The recognition results were then put into a text comparison tool to calculate the error rate against ground truth from manual transcriptions.

The HTR+ model was first transferred from a publicly available model with german historical documents. Because of the bad performance observed, the author rejected the existing training data and tried to train the model with pages of synthetic text and validate it on map sets. The gap between training and validation accuracy (see the learning curve in fig.5.9) became so large to conduct reliable predictions on segmented images. The author further improved the training with more manual labelled text, and the CER has significantly decreased with the increasing amount of training data.

The CRNN+STN trained on Syn90k achieved the best outcomes even without extra training. HTR+, which is also using CRNN structure, shows unsatisfying results. The main influence factor could be the training data used. The training data from the same language at the same time age can not train the model better than a more diverse, much larger volume of English database.

5.2.2 Error Analysis

The confusion matrix records the number of occurrences between characters recognized and the true letters. In figure 5.10, two tables show the confusion matrix of the HTR+ model and the CRNN model. The rows stand for the true letters and the columns display model prediction. The main diagonal from top left to bottom right shows the number of correctly classified letters.

None of the "z" is correctly interpreted by CRNN model. This can be because in the unique font style of Saxon Mile Sheet, "z" is so uniquely written. The top error in CRNN

5 Results and Discussions

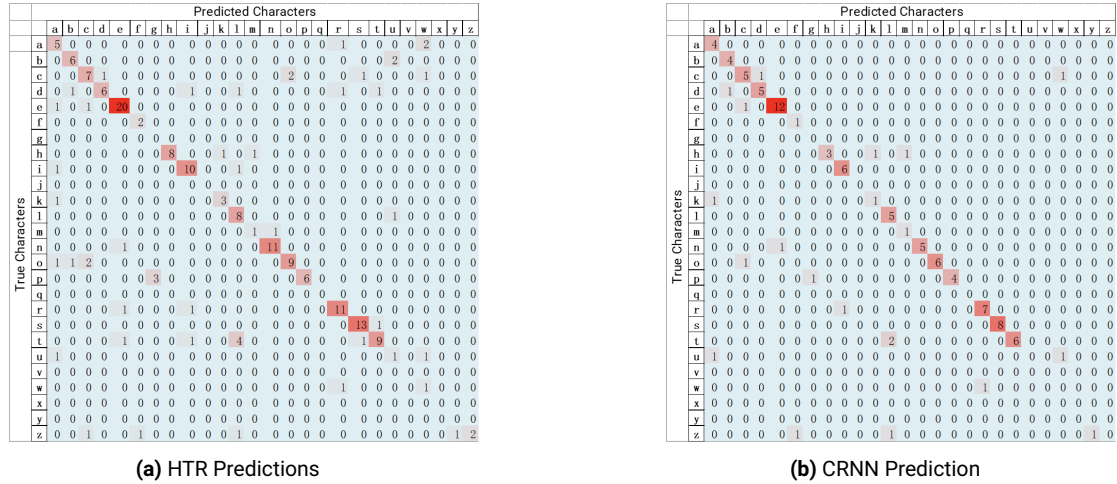


Figure 5.10: Confusion matrix for character recognition. Each row is a frequency count of characters recognized by the system, with darker colors associated with higher frequency counts. The matrix is case-insensitive.

is the "t" and "l" confusion. In HTR+ model, the most common error was "t" and "l" confusion, followed by "p" and "g" confusion.

Letter "t" has a specifically high possibility of being wrongly classified in both models. The vague extraction of "o" and "e" may lead to confusion in both models.

Since only a small number of images were put into the recognition system for testing, the sample size for recognition was not sufficient to show a significant distribution trend. However, this operation can provide direction for subsequently increasing the training samples. For example, in this paper, the recognition of "t" needs to be particularly strengthening the training.

5.3 Overall Performance

Table 5.3 shows the results of detection and recognition of toponyms on a sheet mapping around Dresden Neustadt. From left to right, the table contains the image cropped from detected bounding box, ID, the Universal Transverse Mercator (UTM) coordinates of the center point of image, the Geographic (latitude, longitude) coordinates, the address found for location with Nominatim service, and the recognition results with specifically trained HTR+ model in Transkribus and the trained CRNN model in MMOCR (Recognition2).

The queried place names can serve as an auxiliary assessment of whether the text matches the location, in other words, whether the text box is correctly localized. It can be used to verify the accuracy of the text detection block. Moreover, from another

5.3 Overall Performance

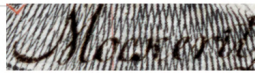

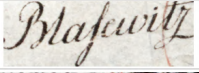



	ID	UTM	latlon	address	HTR+	CRNN
	word_1	(41198	(51.0180953594537, 13.745162075845785	Kleinpestitz/ Mockritz , Zschertnitz, Plauen, Dresden	Bockerwitz	mouhered
	word_2	(41355	(51.02863364008097, 13.76727630737129	128a, Wiener Straße, Seevorstadt-Ost/Großer Garten, Alts	Wrehlen	strehlen
	word_3	(41633	(51.049619547963296, 13.8064412000307	19, Kretschmerstraße, Blasewitz , Dresden, Sachsen, 01	Blasewitz	bolafwng
	word_4	(41169	(51.02002789737003, 13.74105047686047	Räcknitz/Zschertnitz, Zschertnitz, Plauen, Dresden	Kein Gerlin	ppindursa
	word_5	(41225	(51.026328234310014, 13.7487423048190	Räcknitz/Zschertnitz, Zschertnitz , Plauen, Dresden	LnuKe Wdruth	ntlurehnlz
	word_6	(41524	(51.04635719422944, 13.79086641035864	Striesen-West, Striesen , Blasewitz, Dresden	Bausen	struaytn

Table 5.3: Part of the extraction and recognition result of map sheet "Dresden Neustadt"

perspective, the address containing the true place names can compare against the recognition results.

The bold letters in the address column are the true place name in the image. The place names are mostly found on administrative levels as the names of the neighbourhood or suburbs. Two third of the detected word box can be correctly located to the address but not yet correctly transformed to text. Different reasons could cause the missing place names.

First, the change of administrative border makes the point fall on the suburb in the neighbourhood. For example, the third place name "Klein Pestitz" was also not correctly found at the location. It was about a kilometre distant from today's site (see fig.5.11).

Second, geographic bias can be caused by geocoding services. Most of the serving geocoding tools claimed to be accurated down to street address precision. The difference in positional accuracy is discovered by different providers. For example, the second text, "Strehlen", was not found in the queried address. However, by switching the geocoding service to Google API, the name "Strehlen" was found as the sublocality. Third, the place's name has been lost in time. For example, the area "Scheunen" annotated on the map is nowadays only found as a historical building and serves as cafe and culture centre.

For these factors, if a gazetteer or geolocator tool is used to validate or refine the result of toponym detection and recognition, the author would suggest firstly searching with a spatial tolerance and secondly using different service providers.

5 Results and Discussions



Figure 5.11: District "Klein Pestitz" on historical map outlined in green and today's location on OpenStreetMap marked with blue box.

From this integrated table, the performance of text detection and extraction is satisfactory, but the recognition of text still needs much improvement. The existing recognition results are even difficult to perform fuzzy matching for context analysis.

6 Conclusion and Further Work

The thesis reviewed OCR systems and state-of-art deep learning-based techniques (RQ1.1 and RQ2.1). After implementing a proposed deep learning pipeline on Saxon Mile Sheets, the thesis evaluated the practical feasibility of deep learning-based text detection (RQ1.2) and recognition method (RQ2.2) on toponym extraction from a specific type of historical topographic maps (RQ3).

The findings can be summarized as follows:

1. From the literature review, text extraction and recognition algorithms have developed rapidly in recent years. Different genres of methods have been computed and tested in practice. They have proved their effectiveness through testing in various scene text databases. Transfer learning from these pre-trained models would be a good option for the tasks in Cartography as map-specific training data are normally very limited. However, each of these approaches has its own strengths and towards better performance in a specific area under certain contexts. The comparative experiments in this thesis also found that there is a gap between the performance of different pre-trained extraction and recognition models. Direct adoption of trained models in different contexts is not possible. Manual intervention of tuning is still crucial for training. If an unsuitable model is chosen, it will take the researcher much more effort in training to achieve an acceptable result. So, how to choose the suitable method for historical maps from all those approaches is certainly a discussable topic.
2. Synthetic data proved its feasibility of saving efforts for deep learning models without much existing labelled data in text extraction. The extraction of text shape can be comparable to the results trained with a small amount of manually labelled ground truth. However, this approach didn't perform well on the chosen text recognition network. For text transcription, the model still need to be trained on ground truth.
3. The proposed deep learning pipeline proved its efficiency in text localization. Unlike traditional extraction approaches, which are type and size-sensitive, deep learning methods may be able to extract text elements with variants. Compared to the reviewed dl-based OCR work on historical maps, the f1 score and IoU score achieved in this thesis are excellent, with relatively lowest effort spent on training.
4. For existing open-source text recognition tools, recognizing handwritten place names on historical maps with unique font styles is still very challenging. Model performance highly depends on the training data available. Most networks require a large amount of labelled data to achieve a satisfactory prediction. The author

6 Conclusion and Further Work

has not found an effective solution in this paper to free the labour from the heavy manual labelling tasks.

Based on the findings of this paper, the author would like to point out the potential future steps of the study.

1. The data synthesis in the thesis used a straightforward method. The training samples did not simulate the colour shade variations of cursive handwriting. While usability has been demonstrated, extraction in higher precision can be achieved by improving the quality of the synthesized data. A more refined mimic strategy can be first generating a gradient colour space and then using a font mask to extract a text shape with colour variation. The available font generation tools also offer the possibility of randomizing the use of variant letters. Noise and random erasing can be added to imitate the colour fading of scanned historical documents.
2. The large size of the map sheet and complex features set relatively high requirements on hardware. During the processing, the author encountered the run-out-of-memory error many times. The limitation of computing power was solved by reducing computational complexity, for example, cropping the input map into smaller patches. The smaller size of the input image affects the receptive field the feature map can learn from. Cloud computing through web service could be an option to get aggregated supercomputing power. A hypothesis could be tested on whether the efficiency and generalization of models can be improved with a more complex network structure.
3. Gazetteer can be queried based on the localized word box. With the place name database, from one perspective, contextual analysis can be conducted to correct the recognition results. From another perspective, linguistic studies can investigate the changes in place names through history.
4. The experiment, together with polygon and polyline extraction tested by the Historical Semantic Segmentation Team, shows the possibility of developing an integrated system for all feature extraction and interpretation on Saxon Mile Sheets. The toolkit can help automatically translate the raster-format historical map documents to structured geo-data in an effective way. An interface can be developed to let the users input the map and get the required features.

References

- Adelfio, M. D., & Samet, H. (2013). Structured toponym resolution using combined hierarchical place categories [Conference Proceedings]. ACM Press. Retrieved from <https://dx.doi.org/10.1145/2533888.2533931> doi: 10.1145/2533888.2533931
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches [Journal Article]. *arXiv preprint arXiv:1803.01164*.
- Avinash, A. (2018). Hand Written Telugu Character Recognition-Review [Conference Proceedings].
- Barbaresi, A. (2018). Toponyms as Entry Points into a Digital Edition: Mapping Die Fackel [Journal Article]. *Open Information Science*, 2(1), 23-33.
- Beatty, D. M. (2021). Re-inscribing propositions: historic cartography and Philippine claims to the Spratly Islands [Journal Article]. *Territory, Politics, Governance*, 9(3), 434-454. Retrieved from <https://dx.doi.org/10.1080/21622671.2019.1687325> doi: 10.1080/21622671.2019.1687325
- Berchmans, D., & Kumar, S. S. (2014). Optical character recognition: An overview and an insight [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/iccicct.2014.6993174><https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=6993174&ref=> doi: 10.1109/iccicct.2014.6993174
- BILL, R., WALTER, K., & MENDT, J. (2014). Virtuelles Kartenforum 2.0–Verfügbarmachung von Altkarten über eine räumliche Portalanwendung [Journal Article]. *Angewandte Geoinformatik*, 684-693.
- Biswas, S., & Das, A. K. (2012). Text extraction from scanned land map images [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/iciev.2012.6317410> doi: 10.1109/iciev.2012.6317410
- Brunner, H. (2005). *Atlas zur geschichte und landeskunde von sachsen bei heft zu den karten h 12.1 und h 12.2 die sächsische landesaufnahme von 1780 bis 1825 / von hans brunner* [Book]. Verl. der Sächsischen Akad. der Wiss. zu Leipzig : Landesvermessungsamt Sachsen Leipzig ; Dresden: Verl. der Sächsischen Akad. der Wiss. zu Leipzig : Landesvermessungsamt Sachsen. Retrieved from <https://katalog.slub-dresden.de/id/0-1473239567>
- Can, Y. S., & Kabadayi, M. E. (2021). Text Detection and Recognition by using CNNs in the Austro-Hungarian Historical Military Mapping Survey [Conference Proceedings]. ACM. Retrieved from <https://dx.doi.org/10.1145/3476887.3476904> doi: 10.1145/3476887.3476904

References

- Carpenter, K. A., Cohen, D. S., Jarrell, J. T., & Huang, X. (2018). Deep learning and virtual drug screening [Journal Article]. *Future Medicinal Chemistry*, 10(21), 2557-2567. Retrieved from <https://dx.doi.org/10.4155/fmc-2018-0314> doi: 10.4155/fmc-2018-0314
- Chaudhuri, A., Mandaviya, K., Badelia, P., & Ghosh, S. K. (2017). Optical Character Recognition Systems [Book Section]. In A. Chaudhuri, K. Mandaviya, P. Badelia, & S. K Ghosh (Eds.), *Optical character recognition systems for different languages with soft computing* (p. 9-41). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-50252-6_2 doi: 10.1007/978-3-319-50252-6_2
- Chen, C.-H., & DeCurtins, J. L. (1993). Word recognition in a segmentation-free approach to OCR [Conference Proceedings]. In *Proceedings of 2nd international conference on document analysis and recognition (icdar'93)* (p. 573-576). IEEE.
- Chen, Y., Carlinet, E., Chazalon, J., Mallet, C., Duménieu, B., & Perret, J. (2021a). Combining deep learning and mathematical morphology for historical map segmentation [Conference Proceedings]. In *International conference on discrete geometry and mathematical morphology* (p. 79-92). Springer.
- Chen, Y., Carlinet, E., Chazalon, J., Mallet, C., Duménieu, B., & Perret, J. (2021b). Vectorization of Historical Maps Using Deep Edge Filtering and Closed Shape Extraction [Book Section]. In (p. 510-525). Springer International Publishing. Retrieved from https://dx.doi.org/10.1007/978-3-030-86337-1_34 doi: 10.1007/978-3-030-86337-1_34
- Chen, Y.-Y., Huang, W., Wang, W.-H., Juang, J.-Y., Hong, J.-S., Kato, T., & Luyssaert, S. (2019). Reconstructing Taiwan's land cover changes between 1904 and 2015 from historical maps and satellite images [Journal Article]. *Scientific Reports*, 9(1). Retrieved from <https://dx.doi.org/10.1038/s41598-019-40063-1> doi: 10.1038/s41598-019-40063-1
- Chiang, Y.-Y., & Knoblock, C. A. (2011). Recognition of Multi-oriented, Multi-sized, and Curved Text [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/icdar.2011.281> doi: 10.1109/icdar.2011.281
- Chiang, Y.-Y., Leyk, S., & Knoblock, C. A. (2014). A Survey of Digital Map Processing Techniques [Journal Article]. *ACM Computing Surveys*, 47(1), 1-44. Retrieved from <https://dx.doi.org/10.1145/2557423> doi: 10.1145/2557423
- Dunesme, S., Piégay, H., & Mustière, S. (2020). Assessing Historical Maps for Characterizing Fluvial Corridor Changes at a Regional Network Scale [Journal Article]. *Cartographica*, 55(4), 251-265. Retrieved from <https://hal.archives-ouvertes.fr/hal-03371776> (Humanities and Social Sciences/GeographyJournal articles) doi: 10.3138/cart-2019-0025
- Fang, S., Xie, H., Wang, Y., Mao, Z., & Zhang, Y. (2021). Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition [Confer-

- ence Proceedings]. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (p. 7098-7107).
- Grond, M., Brink, W., & Herbst, B. (2016). Text detection in natural images with convolutional neural networks and synthetic training data [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/robomech.2016.7813169><https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=7813169&ref=doi:10.1109/robomech.2016.7813169>
- Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images [Conference Proceedings]. In *Proceedings of the ieee conference on computer vision and pattern recognition* (p. 2315-2324).
- Islam, N., Islam, Z., & Noor, N. (2017). A survey on optical character recognition system [Journal Article]. *arXiv preprint arXiv:1710.05703*.
- Jordan, P. (2009). Some considerations on the function of place names on maps [Conference Proceedings]. In *Proceedings of the 24th international cartographic conference, santiago, chile (icc 2009)* (p. 15-21).
- Kahle, P., Colutto, S., Hackl, G., & Muhlberger, G. (2017). *Transkribus - a service platform for transcription, recognition and retrieval of historical documents* [Book]. doi: 10.1109/ICDAR.2017.307
- Khan, T., Sarkar, R., & Mollah, A. F. (2021). Deep learning approaches to scene text detection: a comprehensive review [Journal Article]. *Artificial Intelligence Review*, 54(5), 3239-3298. Retrieved from <https://dx.doi.org/10.1007/s10462-020-09930-6><https://link.springer.com/content/pdf/10.1007/s10462-020-09930-6.pdf> doi: 10.1007/s10462-020-09930-6
- Khosla, C., & Saini, B. S. (2020). Enhancing Performance of Deep Learning Models with different Data Augmentation Techniques: A Survey [Conference Proceedings]. In *2020 international conference on intelligent engineering and management (iciem)* (p. 79-85). doi: 10.1109/ICIEM48762.2020.9160048
- Kuang, Z., Sun, H., Li, Z., Yue, X., Lin, T. H., Chen, J., ... Lin, D. (2021). MMOCR: A Comprehensive Toolbox for Text Detection, Recognition and Understanding [Journal Article]. *arXiv preprint arXiv:2108.06543*. Retrieved from <https://github.com/open-mmlab/mmocr>
- Laumer, D., Gümgümcü, H., Heitzler, M., & Hurni, L. (2020). A Semi-automatic Label Digitization Workflow for the Siegfried Map [Conference Proceedings]. In *Automatic vectorisation of historical maps, proceedings of the international workshop on automatic vectorisation of historical maps, budapest, hungary* (Vol. 13, p. 55-62).
- Leyk, S., & Boesch, R. (2010). Colors of the past: color image segmentation in historical topographic maps based on homogeneity [Journal Article]. *Geoinformatica*, 14(1), 1-21. Retrieved from <https://dx.doi.org/10.1007/s10707-008-0074-z> doi: 10.1007/s10707-008-0074-z

References

- Li, H., Liu, J., & Zhou, X. (2018). Intelligent Map Reader: A Framework for Topographic Map Understanding With Deep Learning and Gazetteer [Journal Article]. *IEEE Access*, 6, 25363-25376. doi: 10.1109/ACCESS.2018.2823501
- Li, L., Nagy, G., Samal, A., Seth, S., & Xu, Y. (2000). Integrated text and line-art extraction from a topographic map [Journal Article]. *International Journal on Document Analysis and Recognition*, 2(4), 177-185. Retrieved from ocr_text_li
- Li, Z., Chiang, Y.-Y., Tavakkol, S., Shbita, B., Uhl, J. H., Leyk, S., & Knoblock, C. A. (2020). An Automatic Approach for Generating Rich, Linked Geo-Metadata from Historical Map Images [Conference Proceedings]. ACM. Retrieved from <https://dx.doi.org/10.1145/3394486.3403381> doi: 10.1145/3394486.3403381
- Li, Z., Guan, R., Yu, Q., Chiang, Y.-Y., & Knoblock, C. A. (2021). Synthetic Map Generation to Provide Unlimited Training Data for Historical Map Text Detection [Conference Proceedings]. ACM. Retrieved from <https://dx.doi.org/10.1145/3486635.3491070><https://dl.acm.org/doi/pdf/10.1145/3486635.3491070> doi: 10.1145/3486635.3491070
- Lones, M. A. (2021). How to avoid machine learning pitfalls: a guide for academic researchers [Journal Article]. *arXiv preprint arXiv:2108.02497*.
- Long, S., Ruan, J., Zhang, W., He, X., Wu, W., & Yao, C. (2018). TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes [Conference Proceedings]. In *Eccv*.
- Luft, J., & Schiewe, J. (2021). Automatic content-based georeferencing of historical topographic maps [Journal Article]. *Transactions in GIS*, 25(6), 2888-2906. Retrieved from <https://dx.doi.org/10.1111/tgis.12794><https://onlinelibrary.wiley.com/doi/pdfdirect/10.1111/tgis.12794?download=true> doi: 10.1111/tgis.12794
- Lázaro, J., Martín, J. L., Arias, J., Astarloa, A., & Cuadrado, C. (2010). Neuro semantic thresholding using OCR software for high precision OCR applications [Journal Article]. *Image and Vision Computing*, 28(4), 571-578. Retrieved from <https://www.sciencedirect.com/science/article/pii/S026288560900198X><https://www.sciencedirect.com/science/article/pii/S026288560900198X?via%3Dihub> doi: <https://doi.org/10.1016/j.imavis.2009.09.011>
- Löki, V., Schmotzer, A., Takács, A., Süveges, K., Lovas-Kiss, , Lukács, B. A., ... Molnár V, A. (2020). The protected flora of long-established cemeteries in Hungary: Using historical maps in biodiversity conservation [Journal Article]. *Ecology and Evolution*, 10(14), 7497-7508. Retrieved from <https://dx.doi.org/10.1002/ece3.6476><https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7391536/pdf/ECE3-10-7497.pdf> doi: 10.1002/ece3.6476
- Meduri, A., & Goyal, N. (2018). Optical Character Recognition for Sanskrit Using Convolution Neural Networks [Journal Article]. *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, 447-452.
- Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten Optical Character

- Recognition (OCR): A Comprehensive Systematic Literature Review (SLR) [Journal Article]. *IEEE Access*, 8, 142642-142668. Retrieved from <https://dx.doi.org/10.1109/access.2020.3012542https://ieeexplore.ieee.org/ielx7/6287639/8948470/09151144.pdf?tp=&arnumber=9151144&isnumber=8948470&ref=doi:10.1109/access.2020.3012542>
- Mendt, J. (2014). *Virtuelles kartenforum 2.0: Geodateninfrastruktur für die raum-zeit-forschung mit historischen karten* [Book]. SLUB Dresden.
- Michael, J., Weidemann, M., & Labahn, R. (2020). HTR engine based on NNs P3 [Journal Article]. *Horizon*.
- Milleville, K., Verstockt, S., & Van De Weghe, N. (2020). Improving Toponym Recognition Accuracy of Historical Topographic Maps [Conference Proceedings]. Department of Cartography and Geoinformatics ELTE. Retrieved from <https://dx.doi.org/10.21862/avhm2020.08> doi: 10.21862/avhm2020.08
- Müller, M. (2018a). *Aufbereitung der historischen karten* [Pamphlet]. TopMaps Sachsen - Karten von 1780 bis 1810.
- Müller, M. (2018b). *Die meilenblätter und deren inhalt* [Pamphlet]. TopMaps Sachsen - Karten von 1780 bis 1810.
- Nagel, C. (1876). *Die vermessungen im königreiche sachsen: Eine denkschrift mit vorschlägen für eine auf die europäische gradmessung zu gründende rationelle landesvermessung* [Book]. Huhle.
- Nikolenko, S. I. (2021). *Synthetic data for deep learning* (Vol. 174) [Book]. Springer.
- Pezeshk, A., & Tutwiler, R. L. (2010). Extended character defect model for recognition of text from maps [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/ssiai.2010.5483913> doi: 10.1109/ssiai.2010.5483913
- Pezeshk, A., & Tutwiler, R. L. (2011). Automatic Feature Extraction and Text Recognition From Scanned Topographic Maps [Journal Article]. *IEEE Transactions on Geoscience and Remote Sensing*, 49(12), 5047-5063. Retrieved from <https://dx.doi.org/10.1109/tgrs.2011.2157697> doi: 10.1109/tgrs.2011.2157697
- Pouderoux, J., Gonzato, J.-C., Pereira, A., & Guitton, P. (2007). Toponym recognition in scanned color topographic maps [Conference Proceedings]. In *Ninth international conference on document analysis and recognition (icdar 2007)* (Vol. 1, p. 531-535). IEEE.
- Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... Iyengar, S. S. (2019). A Survey on Deep Learning [Journal Article]. *ACM Computing Surveys*, 51(5), 1-36. Retrieved from <https://dx.doi.org/10.1145/3234150https://dl.acm.org/doi/pdf/10.1145/3234150> doi: 10.1145/3234150
- Qin, Y., & Zhang, Z. (2020). Summary of Scene Text Detection and Recognition [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/iciea48937.2020.9248121> doi: 10.1109/iciea48937.2020.9248121
- Reckziegel, M., Wrisley, D. J., Hixson, T. W., & Jänicke, S. (2021). Visual exploration

References

- of historical maps [Journal Article]. *Digital Scholarship in the Humanities*, 36. Retrieved from https://academic.oup.com/dsh/article/36/Supplement_2/ii251/6263475
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation [Book Section]. In (p. 234-241). Springer International Publishing. Retrieved from https://dx.doi.org/10.1007/978-3-319-24574-4_28 doi: 10.1007/978-3-319-24574-4_28
- Rosenblatt, F. (1958). THE PERCEPTRON - A PROBABILISTIC MODEL FOR INFORMATION-STORAGE AND ORGANIZATION IN THE BRAIN [Journal Article]. *Psychological Review*, 65(6), 386-408. Retrieved from <GotoISI>://WOS: A1958WG40900006 (Rosenblatt, f) doi: 10.1037/h0042519
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors [Journal Article]. *Nature*, 323, 533-536.
- Sabbour, N., & Shafait, F. (2013). A segmentation-free approach to Arabic and Urdu OCR [Conference Proceedings]. In *Document recognition and retrieval xx* (Vol. 8658, p. 215-226). SPIE.
- Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions [Journal Article]. *SN Computer Science*, 2(6). Retrieved from <https://dx.doi.org/10.1007/s42979-021-00815-1> <https://link.springer.com/content/pdf/10.1007/s42979-021-00815-1.pdf> doi: 10.1007/s42979-021-00815-1
- Schlegel, I. (2021). Automated Extraction of Labels from Large-Scale Historical Maps [Journal Article]. *AGILE: GIScience Series*, 2, 1-14. Retrieved from <https://dx.doi.org/10.5194/agile-giss-2-12-2021> doi: 10.5194/agile-giss-2-12-2021
- Serra, J., & Vincent, L. (1992). An overview of morphological filtering [Journal Article]. *Circuits Systems and Signal Processing*, 11(1), 47-108. Retrieved from <https://dx.doi.org/10.1007/bf01189221> doi: 10.1007/bf01189221
- Sewak, M. (2019). Introduction to Deep Learning: Enter the World of Modern Machine Learning [Book Section]. In (p. 75-88). doi: 10.1007/978-981-13-8285-7_6
- Sharma, R., Kaushik, B., & Gondhi, N. (2020). Character Recognition using Machine Learning and Deep Learning - A Survey [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/esci48226.2020.9167649> doi: 10.1109/esci48226.2020.9167649
- Shi, B., Bai, X., & Belongie, S. (2017). Detecting Oriented Text in Natural Images by Linking Segments [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/cvpr.2017.371> doi: 10.1109/cvpr.2017.371
- Shi, B., Wang, X., Lyu, P., Yao, C., & Bai, X. (2016). Robust Scene Text Recognition with Automatic Rectification [Journal Article]. Retrieved from <https://dx.doi.org/10.48550/arxiv.1603.03915> doi: 10.48550/arxiv.1603.03915
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale

- image recognition [Journal Article]. *arXiv preprint arXiv:1409.1556*.
- Springenberg, J., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for Simplicity: The All Convolutional Net [Journal Article].
- Stams, M., & Stams, W. (1981). *Die große topographische landesaufnahme von sachsen 1780 bis 1811 und ihre folgekarten* [Book]. Sektion Geodäsie u. Kartogr. d. Techn. Univ., Kartogr.-techn. Einrichtung.
- Ståhl, N., & Weimann, L. (2022). Identifying wetland areas in historical maps using deep convolutional neural networks [Journal Article]. *Ecological Informatics*, 68, 101557. Retrieved from <https://dx.doi.org/10.1016/j.ecoinf.2022.101557> doi: 10.1016/j.ecoinf.2022.101557
- Sánchez-Ramírez, E., Segovia-Hernández, J. G., & Hernández-Vargas, E. A. (2020). Artificial Neural Network to capture the Dynamics of a Dividing Wall Column [Book Section]. In S. Pierucci, F. Manenti, G. L. Bozzano, & D. Manca (Eds.), *Computer aided chemical engineering* (Vol. 48, p. 133-138). Elsevier. Retrieved from <https://www.sciencedirect.com/science/article/pii/B9780128233771500239> doi: <https://doi.org/10.1016/B978-0-12-823377-1.50023-9>
- Tavakkol, S., Chiang, Y.-Y., Waters, T., Han, F., Prasad, K., & Kiveris, R. (2019). Kartta Labs: Unrendering Historical Maps [Conference Proceedings]. ACM. Retrieved from <https://dx.doi.org/10.1145/3356471.3365236> doi: 10.1145/3356471.3365236
- Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. (2016). Detecting Text in Natural Image with Connectionist Text Proposal Network [Book Section]. In (p. 56-72). Springer International Publishing. Retrieved from https://dx.doi.org/10.1007/978-3-319-46484-8_4 doi: 10.1007/978-3-319-46484-8_4
- Uhl, J. H., Leyk, S., Li, Z., Duan, W., Shbita, B., Chiang, Y.-Y., & Knoblock, C. A. (2021). Combining Remote-Sensing-Derived Data and Historical Maps for Long-Term Back-Casting of Urban Extents [Journal Article]. *Remote Sensing*, 13(18), 3672. Retrieved from <https://dx.doi.org/10.3390/rs13183672> <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8691741/pdf/nihms-1761682.pdf> doi: 10.3390/rs13183672
- Velázquez, A., & Levachkine, S. (2004). Text/Graphics Separation and Recognition in Raster-Scanned Color Cartographic Maps [Book Section]. In (p. 63-74). Springer Berlin Heidelberg. Retrieved from https://dx.doi.org/10.1007/978-3-540-25977-0_6 doi: 10.1007/978-3-540-25977-0_6
- Walz, U., & Berger, A. (2003). Georeferenzierung und Mosaikerstellung historischer Kartenwerke-Grundlage für digitale Zeitreihen zur Landschaftsanalyse [Journal Article].
- Walz, U., & Schumacher, U. (2011). Sächsische Meilenblätter als Quelle der Kulturlandschaftsforschung am Beispiel der Sächsischen Schweiz [Journal Article]. *Cartographica Helvetica*(44), 3.
- Wang, H., & Raj, B. (2017). On the origin of deep learning [Journal Article]. *arXiv preprint arXiv:1702.07800*.

References

- Weinman, J. (2013). Toponym Recognition in Historical Maps by Gazetteer Alignment [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/icdar.2013.209> doi: 10.1109/icdar.2013.209
- Weinman, J., Chen, Z., Gafford, B., Gifford, N., Lamsal, A., & Niehus-Staab, L. (2019). Deep Neural Networks for Text Detection and Recognition in Historical Maps [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/icdar.2019.00149> doi: 10.1109/icdar.2019.00149
- Wu, S., Heitzler, M., & Hurni, L. (2022). Leveraging uncertainty estimation and spatial pyramid pooling for extracting hydrological features from scanned historical topographic maps [Journal Article]. *GIScience Remote Sensing*, 59(1), 200-214. Retrieved from <https://dx.doi.org/10.1080/15481603.2021.2023840><https://www.tandfonline.com/doi/full/10.1080/15481603.2021.2023840> doi: 10.1080/15481603.2021.2023840
- Yamada, H., Yamamoto, K., & Hosokawa, K. (1993). Directional mathematical morphology and reformatized Hough transformation for the analysis of topographic maps [Journal Article]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4), 380-387. Retrieved from <https://dx.doi.org/10.1109/34.206957> doi: 10.1109/34.206957
- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology [Journal Article]. *Insights into Imaging*, 9(4), 611-629. Retrieved from <https://dx.doi.org/10.1007/s13244-018-0639-9> doi: 10.1007/s13244-018-0639-9
- Zhang, S.-X., Zhu, X., Hou, J.-B., Liu, C., Yang, C., Wang, H., & Yin, X.-C. (2020). Deep Relational Reasoning Graph Network for Arbitrary Shape Text Detection [Conference Proceedings]. IEEE. Retrieved from <https://dx.doi.org/10.1109/cvpr42600.2020.00972> doi: 10.1109/cvpr42600.2020.00972
- Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., & Zhang, W. (2021). Fourier contour embedding for arbitrary-shaped text detection [Conference Proceedings]. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 3123-3131).
- Zimmermann, G. (2017). Neue Perspektiven für historische Karten [Journal Article]. *BIS - Das Magazin der Bibliotheken in Sachsen - Jg. 10. 2017, H. 1*. Retrieved from <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-79329>