FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Automated Semantic Annotation of Historical Catalogues

MASTERARBEIT

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Visual Computing

eingereicht von

David Körner
Matrikelnummer 00725733

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Mitwirkung: Markus Diem
Florian Kleber

Wien, 26. Juni 2020

David Körner

Robert Sablatnig

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Automated Semantic Annotation of Historical Catalogues

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Visual Computing

by

David Körner
Registration Number 00725733

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Assistance: Markus Diem
Florian Kleber

Vienna, 26th June, 2020

David Körner | Robert Sablatnig

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

David Körner

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 26. Juni 2020

David Körner

Danksagung

Ich möchte mich an dieser Stelle bei allen Menschen bedanken, die mich im Laufe meines Studiums begleitet und unterstützt haben. Jeder einzelne hat seinen Teil dazu beigetragen, dass ich diese Arbeit schreiben und damit den Abschluss meines Studiums erreichen konnte.

Ein ganz besonderer Dank gilt meiner Familie, die mir das Studium überhaupt erst ermöglicht hat und auch in jeder schwierigen Situation stets hinter mir gestanden ist.

Ich widme diese Arbeit meinen Eltern, die trotz so manch schwieriger Zeiten immer für mich da waren.

Danke!

Acknowledgements

I would like to thank my advisors Florian Kleber and Markus Diem for their advice, feedback and most of all their patience. I would also like to thank the PRImA Research Lab team for providing the data set and evaluation tools of ICDAR2013 Competition on Historical Book Recognition. Finally, I would also like to thank Christina Bartosch and her team for providing access to the catalogues data sets of the project *Exhibitions of Modern European Painting 1905-1915* (funded by the Austrian Science Fund FWF, Project Number P 29997-G24) and their support.

Kurzfassung

Historische Dokumente enthalten verschiedenste Arten von Informationen, die zum besseren Verständnis bestimmter Zeitabschnitte der Geschichte genutzt werden können. Kunstausstellungskataloge stellen eine spezielle Art von historischen Dokumenten dar, die wertvolle Informationen über die Kunstgeschichte enthalten. Das Forschungsprojekt *Ausstellungen moderner europäischer Malerei 1905-1915* der Universität Wien bemüht sich um die Sammlung und Digitalisierung von Kunstausstellungskatalogen, um die Geschichte der modernen Malerei zu erforschen. Das Projekt befasst sich mit einer Sammlung von mehr als 1300 Katalogen. Die manuelle Digitalisierung dieser Sammlung ist ein aufwendiger Prozess, selbst wenn zusätzliche Software wie Tesseract verwendet wird. In dieser Arbeit wird ein automatisiertes System für die Extraktion von spezifischen Informationen vorgestellt. Das System beschränkt sich auf die Sammlung von Ausstellungskatalogen und vereinfacht den Digitalisierungsprozess in Kombination mit Tesseract.

Der erste Schritt des Systems ist eine Seitensegmentierung. Zu diesem Zweck wird ein Ansatz basierend auf *Maximally Stable Extremal Regions* und eine anschließende Gruppierung der Textregionen verwendet. Die dabei entstandenen Textbereiche werden durch Anwendung einer Fontklassifikation auf Wortebene weiter verfeinert. Diese Klassifizierung erfolgt mittels einer Texturanalyse der Wortregionen basierend auf Gabor-Filterung. Die dadurch erlangten Fontinformationen werden dann verwendet, um bestimmte Kategorien von Informationen zu identifizieren, die sich durch eindeutige Fontstile unterscheiden. Schließlich ist es durch die Kombination dieser Schritte mit der optischen Texterkennung von Tesseract möglich, automatisiert verschiedene Kategorien von Informationen aus den Katalogen zu extrahieren.

Die vorgeschlagene Methode zur Seitensegmentierung wird anhand des Datensatzes der *ICDAR2013 Competition on Historical Book Recognition* evaluiert und ist in der Lage, die Segmentierungsergebnisse von Tesseract zu übertreffen. Darüber hinaus wird der Gabor-Filteransatz, der für die Klassifizierung von Fonts verwendet wird, anhand unterschiedlicher Ausstellungskataloge evaluiert und erreicht eine Erkennungsrate von über 90% für zugeschnittene Wortbilder. Durch die Verwendung der vorgeschlagenen Schritte in Kombination mit der Texterkennung von Tesseract ist es möglich, die digitale Erfassung der Ausstellungskataloge zu erleichtern und den manuellen Aufwand im Digitalisierungsprozess zu reduzieren.

Abstract

Historical documents comprise all kind of information that can be used to gain knowledge about certain periods in time. Art exhibition catalogues represent a special type of historical documents that contains information valuable for the research on the history of art. The research project *Exhibitions of Modern European Painting 1905-1915* at the University of Vienna thrives to gather and digitize art exhibition catalogues in order to perform research on the history of modern painting. The project deals with a collection of more than 1300 catalogues. The manual digitization of this collection is a cumbersome process even when utilizing additional state-of-the-art software like Tesseract. In this thesis an automated system for the extraction of specific information is proposed. The system is limited to the collection of exhibition catalogues and provides means to improve the digitization process in combination with Tesseract.

The first step of the system is a page segmentation. For this purpose, an approach based on *Maximally Stable Extremal Regions* and a subsequent text region grouping is used. The resulting text regions are then further refined by applying a word level font style classification. This classification is done using a texture analysis of the word regions based on Gabor filtering. The computed font style information of text regions is then utilized in order to identify specific categories of information that are formatted using a unique font style. Finally, by combining these steps with the optical character recognition methodology of Tesseract it is possible to automatically extract different categories of information from the catalogues.

The proposed page segmentation methodology is evaluated on the data set of the *ICDAR2013 Competition on Historical Book Recognition* and is able to outperform the segmentation results of Tesseract. In addition, the proposed Gabor filtering approach used for font style classification is evaluated using varying exhibition catalogues and achieves recognition rates above 90% for cropped word images. By using the proposed stages in combination with the optical character recognition of Tesseract it is possible to ease the recognition of the exhibition catalogues and reduce the need for manual effort in the digitization process.

Contents

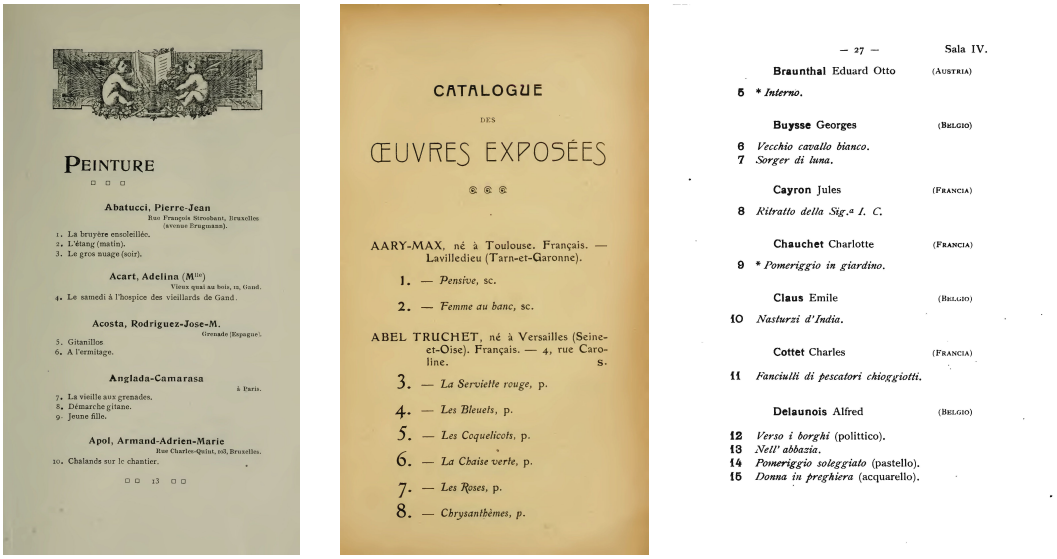
CHAPTER 1

Introduction

Historical documents such as registers of birth, newspapers and others comprise information that can be used to facilitate historical research. In order to access the information contained in these documents they are digitized and subsequently their content is analysed. This process can also be used to process large collections of documents e.g. from libraries. This work deals with a collection of historical catalogues that originate from varying sources. The catalogues are collected as part of the project *Exhibitions of Modern European Painting 1905-1915* (funded by the Austrian Science Fund FWF, Project Number P 29997-G24) [art] at the University of Vienna. In this project the history of modern European painting in the period from 1905-1915 is studied based on art exhibitions. By extracting and analysing data records contained in the art exhibition catalogues the knowledge base is expanded and new insights are gained.

In order to ease the process of extracting information from the art exhibition catalogues the use of text recognition software, like Tesseract [tes], Abby FineReader [Fre], and others is considered. This type of software can be used for an automatic detection and recognition of text regions. However, text regions also need to be classified according to the specific information they represent. The state-of-the-art text recognition solutions are not sufficient for this task since they are limited with respect to their abilities to deal with historical documents and to perform semantic labelling. In Figure 1.1 images of pages from three different exhibition catalogues are shown. The figure shows that the exhibition catalogues contain specific types of information such as name and address of an artist, or title and identification number of a painting. This means that despite the use of text recognition a manual classification of the extracted text is needed to convey its semantic interpretation into a database that can be used for further research.

This work deals with the aforementioned collection of historical art exhibition catalogues and thrives to automate the recognition process of the contained information. For this purpose, input document images are analysed with respect to their layout and the semantic interpretation of the detected text regions is determined. Based on the



PEINTURE

Abatucci, Pierre-Jean
Rue François Stroobant, Bruxelles
(avenue Brugmann).
1. La bruyère ensoleillée.
2. L'étang (matin).
3. Le gros nuage (soir).

Acart, Adelina (Mlle)
Vieux quai au bois, 11, Gand.
4. Le samedi à l'hospice des vieillards de Gand.

Acosta, Rodriguez-Jose-M.
Grenade (Espagne).
5. Gitanillos
6. A l'ermitage.

Anglada-Camarasa
à Paris.
7. La vieille aux grenades.
8. Démarche gitane.
9. Jeune fille.

Apol, Armand-Adrien-Marie
Rue Charles-Quint, 108, Bruxelles.
10. Chalands sur le chantier.

13

CATALOGUE
DES
ŒUVRES EXPOSÉES

AARY-MAX, né à Toulouse. Français. — Lavilledieu (Tarn-et-Garonne).
1. — *Pensive*, sc.
2. — *Femme au banc*, sc.

ABEL TRUCHET, né à Versailles (Seine-et-Oise). Français. — 4, rue Caroline. S.
3. — *La Serviette rouge*, p.
4. — *Les Bleuets*, p.
5. — *Les Coquelicots*, p.
6. — *La Chaise verte*, p.
7. — *Les Roses*, p.
8. — *Chrysanthèmes*, p.

— 27 —
Sala IV.

Braunthal Eduard Otto (Austria)
5 * *Interno.*

Buysse Georges (Belgio)
6 *Vecchio cavallo bianco.*
7 *Sorger di luna.*

Cayron Jules (Francia)
8 *Ritratto della Sig.a I. C.*

Chauchet Charlotte (Francia)
9 * *Pomeriggio in giardino.*

Claus Emile (Belgio)
10 *Nasturzi d'India.*

Cottet Charles (Francia)
11 *Fanciulli di pescatori chioggiotti.*

Delaunois Alfred (Belgio)
12 *Verso i borghi* (polittico).
13 *Nell' abbazia.*
14 *Pomeriggio soleggiato* (pastello).
15 *Donna in preghiera* (acquarello).

(a) (b) (c)

Figure 1.1: Images showing samples of pages from three different art exhibition catalogues. The catalogues are (a) Exposition générale des Beaux-Arts - Bruxelles 1907 [Arcc] (b) Salon d'automne - Exposition de 1907 - Paris [Arcb] (c) VI. Esposizione d'Arte della Città di Venezia [Arca].

extracted semantic annotation recognized text can be classified without the need of manual intervention.

1.1 Motivation

The analysis of the historical art exhibition catalogues can be used to identify chronology, geography and networks of artists and their respective work. For this purpose, it is necessary to extract information from catalogues of exhibitions in different institutions and cities within a specific time period. In the project *Exhibitions of Modern European Painting 1905-1915* [art] more than 1300 art exhibitions held in the period 1905-1915 are analysed. The acquisition of the information contained in these catalogues is a tedious and time-consuming process. Therefore, automating the digitization process of the catalogues can speed up the process and reduce the human effort.

The exhibition catalogues contain lists that comprise information about artists and the artworks they presented at the exhibitions. This means that the catalogues contain specific information such as name and address of artist, and also identification numbers and names of artworks. The semantic interpretation of these different categories of information is indicated by formatting properties of the font and sometimes also by geometrical properties such as the position. In some cases, different categories of information contained in the same text line can only be discriminated based on their font

style. The additional extraction of this semantic interpretation adds value to the textual data of the catalogues. Text that is categorized according to its semantic interpretation can be directly added to a database and used to facilitate further research.

In order to automate the digitization of the textual data and annotate it according to its semantic interpretation a specialized document analysis system is required. As shown in the ICDAR2013 Competition on Historical Book Recognition (HBR2013) [ACPP13a] the state-of-the-art systems ABBYY FineReader [Fre] and Tesseract [tes] are outperformed in terms of page segmentation on historical documents. In addition, both systems are limited in their abilities to extract font style information, especially in historical documents, because they are designed to analyse paragraphs or entire pages rather then individual words. Hence, in this thesis a system based on page segmentation and font style classification adapted to the exhibition catalogues is proposed. The aim of the system is to provide a page segmentation approach that is able to deal with the specific layout of the catalogues and use the unique formatting properties to extract additional semantic information. In addition, the system is designed to be able to incorporate the Tesseract engine for the providing text recognition results.

1.1.1 Scope of Discussion

This thesis is focused on the layout analysis of the historical art exhibition catalogues described earlier in this section. The aim is to use additional layout information to improve the abilities of state-of-the-art Optical Character Recognition (OCR) systems to automatically extract specific information. Therefore, the scientific research question is formulated as follows: *Given the knowledge about repeated formatting of specific categories of information in a catalogue collection. Is it possible to incorporate the extraction of formatting characteristics into a document analysis system such that it outperforms generic state-of-the-art systems like Tesseract in terms of extracting the specific information?*

The development of the system is targeted at the discussed collection of catalogues. This means that the input document images are expected to be machine printed historical documents containing only Latin scripts. In addition, the basic assumption is that the documents originate from different digitization projects that used a systematic image acquisition setup. Therefore, it is assumed that the input documents images are rotated such that the main text orientation is aligned horizontally and do not contain severe image artefacts caused by insufficient lighting, specular highlights, etc.

Moreover, the development of an own text recognition approach is considered beyond the scope of this work. The proposed system is focused on the task of layout analysis and combined with the OCR methodology provided by the freely available Tesseract OCR engine [tes].

1.1.2 Aim of the Thesis

The aim of the thesis is to provide a system that is able to automate the digitization process of the historical art exhibition catalogues. Applying the Tesseract OCR engine that also includes a layout analysis stage is insufficient for the segmentation and semantic annotation of the text regions. Therefore, a Document Image Analysis (DIA) approach based on state-of-the-art methods in page segmentation and font style classification is developed and adapted to the specific requirements of the application. In addition, the system is designed such that it can be combined with the Tesseract engine in order to provide text recognition results.

The page segmentation needs to be adapted to the specific layout of the catalogues and should be able to deal with the noise and varying backgrounds contained in historical document images. Therefore, the page segmentation is based on state-of-the-art methods presented in the page segmentation competition series [ACPP11] held at the International Conference on Document Analysis and Recognition (ICDAR) and designed to be robust to artefacts. In addition, the approach is adapted to historical documents by evaluating it on historical documents.

The font style classification is used to extract formatting properties in the catalogues. This means that the classification needs to be performed on word level in order to identify variations of the font style in a text line. In addition, the method also needs to be able to deal with the challenges of historical documents. Hence, a robust method based on texture analysis is used and evaluated on the catalogue images.

Finally, the results of the previous stages need to be combined with each other. This means that the page segmentation results need to be adapted in order to pass them to the font style classification stage. For this purpose, the extracted text lines need to be segmented into words that are then annotated according the font style classification results.

1.1.3 Main Contribution

The main contribution of this thesis is the assessment of the performance and general characteristics of a document analysis system targeting the extraction of specific information from historical catalogues. The actual information extraction process is based on the knowledge about formatting characteristics of the targeted documents. Therefore, a research on state-of-the-art methods in DIA with a focus on page segmentation, optical font recognition and text recognition is done. Based on this research a DIA system combining page segmentation and font style classification is developed.

The page segmentation method is based on state-of-the-art methods and designed with respect to the domain of historical documents. For the evaluation of the approach a data set focused on historical books is used and a comparison with state-of-the-art methods is done. In addition, the segmentation approach is also adapted to the targeted catalogue data sets and segmentation results are presented. For the font style classification a word-

level approach based on texture analysis is developed and adapted to the catalogue data. The evaluation of the approach is done using synthetic and real-world (i.e. catalogue pages) data in order to provide a detailed analysis of the performance and gain insights on general characteristics.

Finally, the combination of both stages is described and evaluated on the catalogue data sets. This process represents the identification of specific information contained in the catalogues based on their formatting characteristics. Additionally, the integration of text recognition software into the proposed system is explained in order to extract the textual content of the identified regions.

1.2 Evaluation and Results

The proposed page segmentation method is evaluated on the HBR2013 test data set. This competition is focused on historical books and provides a comparative evaluation of state-of-the-art methods. This edition of the competition most closely resembles the input data that is targeted in this work. Therefore, the proposed page segmentation approach is compared against the results of this competition and the detailed results are used to assess its characteristics. The results show that the proposed segmentation method achieves an overall success rate of 75,8% and outperforms Tesseract 3 [tes] by 8,6%. The proposed method performs better on challenging documents with complex layouts and varying background. However, the method lacks a non-text filtering approach and therefore suffers from misclassification of graphical regions and noise. Therefore, the method is not able to provide competitive results compared to the other methods evaluated in the competition. However, the missing non-text filtering does not affect the segmentation of the catalogue images since they contain only text. Segmentation results on the catalogues show that the text region grouping process of the proposed approach is efficient for the extraction of text lines. The approach is able to deal with the sparse layout of the catalogues and results only in minor errors caused by oversegmentation.

For the evaluation of the font style classification approach synthetic and catalogue images are used. The synthetic data is used to assess the characteristics of the approach and evaluate specific aspects that cannot be tested with the limited catalogue data. The evaluation shows that the performance of the approach is improved by adapting the parameters of the used Gabor filter bank to the application domain. In addition, recommendations for the choice of the classifier, the minimum amount of training data and the texture generation process are presented. The proposed approach achieves word level recognition rates above 90% for the discrimination of font styles characteristics such as style (i.e. regular, bold, italic), typeface and size. However, the results also reveal that the performance depends on the word length. Based on a word length evaluation it is shown that short words with less than five characters result in a clearly lower performance dropping from an average of 91.7% to 83.1% for short words.

The combination of page segmentation and font style classification is evaluated by using the extracted text regions as input for the font style classification. The results show that

the classification performance is slightly reduced due to segmentation errors of the page segmentation stage. However, using the combination of both methods it is still possible to maintain recognition rates above 90% for the majority of the trained font style classes. Therefore, the extracted and classified text regions can be processed with Tesseract in order to provide labelled textual fragments that represent specific information extracted from the catalogues.

1.3 Structure of the Thesis

The remainder of this work is structured as follows. In Chapter 2 related work from the field of DIA is presented in four sections. A brief overview of document layout analysis is presented in Section 2.1. In the Sections 2.2 and 2.3 state-of-the-art methods for the tasks of page segmentation and optical font recognition are presented. Finally, Section 2.4 gives an introduction to OCR and how it can be integrated into document layout analysis systems.

In Chapter 3 the methodology used to build the proposed recognition system is described. The chapter is divided into two sections representing the major components of the system. Section 3.1 presents the details on the page segmentation that is used extract text region candidates and group them together. Thereafter, in Section 3.2 the approach used to perform font style classification is described. The latter section provides details on the extraction of Gabor features and how they are used for the classification of font styles.

The evaluation results for the components of the proposed system are presented in Chapter 4. The results of the page segmentation on the HBR2013 test data set and detailed results on the individual stages of the approach are presented in Section 4.1. The detailed evaluation of the font style classification process including results on the historical catalogue document images is shown in Section 4.2. In Section 4.1.3 evaluation results for the combination of the two previous stages are presented.

Finally, in Chapter 5 the work is concluded and an outlook including potential improvements as well as extensions to the proposed system is presented.

CHAPTER 2

State-of-the-Art

The annotation system proposed in this work is based on methods from the field of DIA. The three main parts of the system are page segmentation, Optical Font Recognition (OFR) and OCR. The first two parts both represent document layout analysis tasks whereas the OCR recognizes the text within the extracted regions. In this chapter state-of-the-art methods in DIA for the aforementioned three tasks are discussed. The discussion is focused on methods used on historical printed documents which is the application domain of the proposed system.

Section 2.1 gives a general overview of document layout analysis. In the following sections the subtasks of page segmentation and OFR are discussed in more detail. Section 2.2 presents related work in page segmentation and Section 2.3 gives an overview of state-of-the-art OFR methods. Finally, Section 2.4 gives a brief overview about OCR.

2.1 Document Layout Analysis

The extraction of the document layout is a major part of the proposed DIA system. The individual stages of such a system are usually adapted to the specific application domain but they share a similar workflow. For example, Eskenazi et al. [EGKO17] present an overview of a *classic* DIA system for offline document image segmentation and recognition which is shown in Figure 2.1. The figure shows that the first stage after the image acquisition is the preprocessing, to remove noise, deskew the image, or apply other image enhancements. The next stage is the extraction of the layout information which is divided into segmentation and classification of the preprocessed input images. Finally, in the last part of the *classic* system the extracted layout information is passed on and used to perform tasks such as indexing, OCR and others. In this section a short overview of the field of layout analysis with a focus on printed historical document is presented.

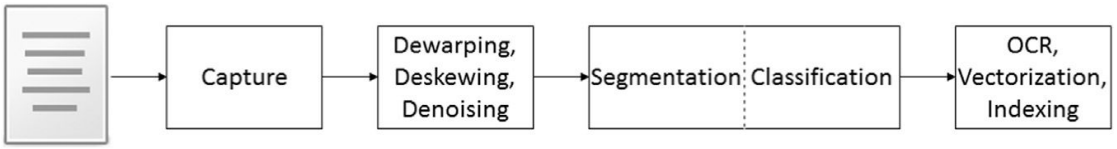


Figure 2.1: Exemplary workflow of a layout analysis system, courtesy by [EGKO17].

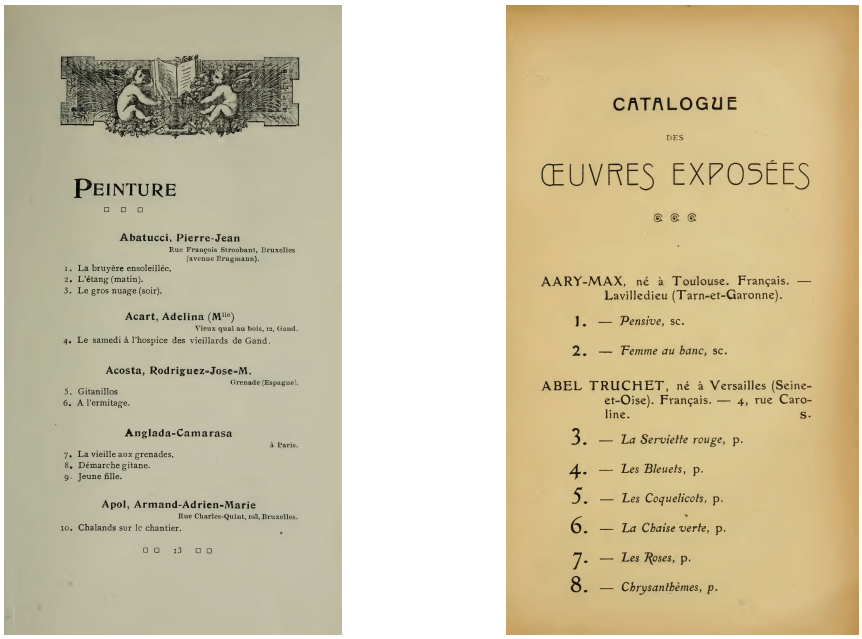


Figure 2.2: Images showing samples of pages from different art exhibition catalogues, courtesy by [Arcc, Arcb].

Document layout analysis can be divided into geometrical and logical layout analysis. The goal of geometrical layout analysis is to segment a document into regions of varying type such as text, illustrations, tables and others. Whereas, logical layout analysis aims to segment text regions according to their logical role (e.g. title, body, foot notes, etc.).

In the proposed system a page segmentation stage is used to extract text regions. This stage represents the extraction of the geometrical layout. Then an OFR method is used to determine the font style which is used for the extraction of the logical layout. Examples of catalogue images the proposed system is applied to are shown in Figure 2.2.

As stated by Eskenazi et al. [EGKO17] logical layout analysis methods are strongly dependent on the type, and layout of the processed documents. Therefore, the methods usually cannot be generalized to other documents. This also means that different logical layout analysis methodologies are specific to a certain document type. Hence, the discussion of the logical layout analysis in Section 2.3 is limited to the task of font recognition and documents that can be segmented based on the contained font styles.

A typology of layout analysis methods can be defined by categorizing them according to the characteristics of the data they are applied to. Clausner et al. [CAP17] state that layout analysis methods are frequently adapted according to their application. As already mentioned, the focus in this section is on printed historical documents and more precisely on documents containing Latin scripts. This type of document images comprises challenges such as degradation artefacts, bleed-through text and others. The processed catalogue images only exhibit minor artefacts and therefore no specific preprocessing steps dealing with image artefacts are applied. The main challenge is the extraction of layout information.

Binarization is a common preprocessing step used in several state-of-the-art methods [CYL13, TNK16] in layout analysis before the extraction of foreground (text) components. In order to deal with historical documents robust binarization methods that are able to deal with local variations are required. A detailed evaluation of state-of-the-art binarization methods is presented by the Document Image Binarization Competition (DIBCO) series [PZKG18].

Scanned document image might be skewed due to inaccuracies or errors in the image acquisition process. For this purpose, skew detection methods based on techniques like *projection profiles* [Bai95], and *Hough transform* [AF00] can be adopted in the preprocessing stage. In the proposed system no additional skew correction is applied, since it is assumed that the images are correctly aligned, but the page segmentation is designed to be robust against minor inaccuracies in the text alignment.

Document images may exhibit varying types of layouts and additional challenges such as page curl, irregular spacing and marginal notes. A schematic illustration of different classes of layouts is given in Figure 2.3. Kise et al. [Kis14] describes the following layout classes: *rectangular* a), *Manhattan* b), *non-Manhattan* c) and *overlapping* d) + e). Rectangular layouts consist only of non-overlapping, rectangular regions having boundaries that are parallel or perpendicular to the page borders. Similarly, Manhattan layouts consist of regions having only boundaries that are parallel or perpendicular to each other. Whereas, non-Manhattan layouts may also include regions with slanted boundaries, or in case of overlapping layouts even interleaved regions. For the latter class Figure 2.3 shows two different examples, in d) a text region contained within an image, and in e) two intersecting text regions are illustrated. The limitations of layout analysis methods to deal with these different layout classes is discussed in more detail in Section 2.2.

A way to deal with local variations in historical documents is to use local features descriptors. This allows to extract candidate text regions without previous binarization. Garz et al. [GSD11] use Scale Invariant Feature Transform (SIFT) descriptors [Low04] on historical manuscript images and Wang et al. [WFS+15] propose a robust text line extraction method based on Maximally Stable Extremal Region (MSER).

MSER proposed by Matas et al. [MCUP04], originally designed for robust stereo matching, is a region detector based on repeated thresholding operations. Due to the robustness

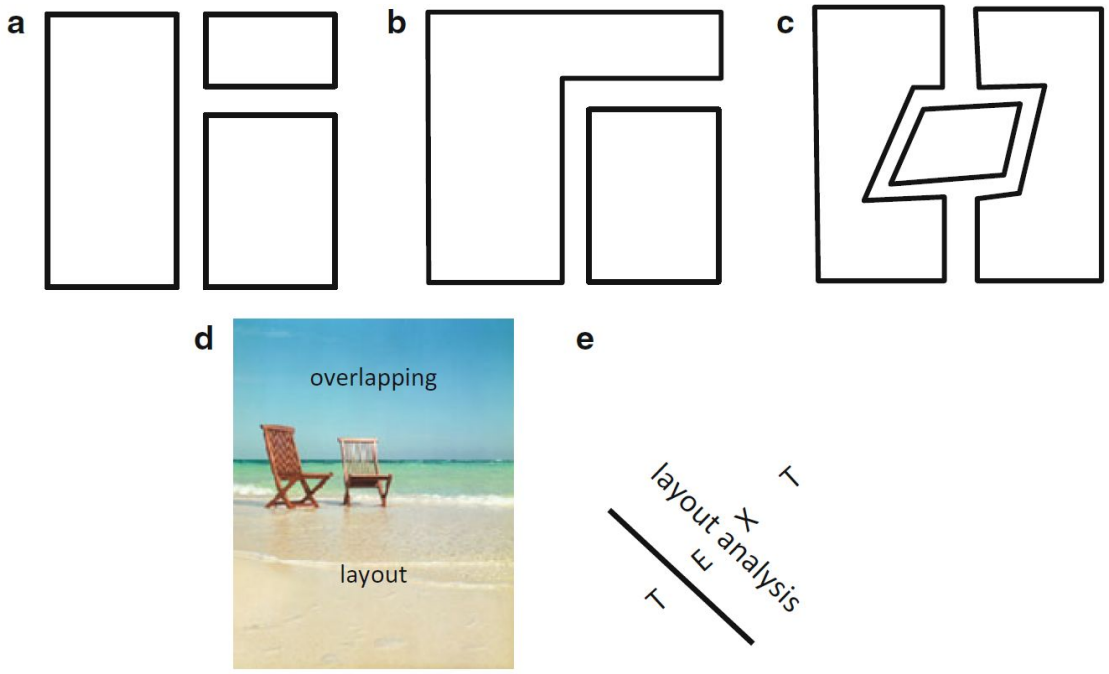


Figure 2.3: Different types of document layouts: a) Rectangular, b) Manhatten, c) non-Manhatten, d) and e) overlapping. Figures courtesy by Kise et al. [Kis14].

of MSER it is also used in scene text detection applications [NM12]. A drawback of the method is the large number of extracted regions since the result of the method is a component tree representing a set of nested extremal regions. Therefore, additional pruning strategies are required to reduce the number of detected regions. SIFT on the other hand produces a set of interest points based on local gradient features. Text regions are represented by multiple interest points. This means that text regions need to be determined by employing an additional grouping process based on local information.

As already mentioned, authors usually adapt their layout analysis methods according to the application scenario it is used in. Therefore, there is a need for an objective comparison for layout analysis methods on a common contemporary data set. In order to achieve this goal a biennial page segmentation competition series [ACPP11] is held at the ICDAR. The competition uses varying data sets and since the 6th edition of the competition [ACPP11] an improved scenario driven evaluation scheme is adopted. The best performing methods presented in the competition series are presented in more detail in the following section.

Furthermore, the competitions in 2011[ACPP11] and 2013[ACPP13a, ACPP13b] are emphasised on historical documents to account for the large number of projects on library digitisations. The data set used for the HBR2013 [ACPP13a] most closely resembles the input data used for this thesis. Therefore, the proposed page segmentation approach is compared against the results of this competition.

2.2 Page Segmentation

Page segmentation methods used in layout analysis methods can be grouped according to the order of processing as stated by Namboodiri and Jain [NJ07]. Following their typology methods are classified as bottom-up, top-down or hybrid. Bottom-up methods start with primitives (e.g. pixels, Connected Component (CC), patches, etc.) and group them together to form higher-level structures such as words, lines, paragraphs, or other types of zones. In contrast to this top-down methods start with the image of a whole page and utilize knowledge about the document layout to iteratively divide the page into smaller sub-regions. A combination of bottom-up grouping processes and top-down splitting of regions is used in hybrid methods.

Another possibility for categorizing segmentation methods is to consider limitations in terms of the ability to deal with different kind of document layouts or types [NJ07, Kis14]. In this section methods are grouped according to their processing order but in addition their ability to deal with the specific class of printed historical documents is analysed. As pointed out by Antonacopoulos et al. [ACPP13a, ACPP13b] there is a convergence in the methodologies used in the segmentation competitions organised at the ICDAR 2013. These approaches are detailed in the section on hybrid methods.

2.2.1 Top-down Methods

Some of the earliest methods for page segmentation are based on top-down processing of pages. The Run-Length Smearing Algorithm (RLSA) by Wong et al. [WCW82] assumes that input images are binary and skew is corrected in a preprocessing step. The algorithm uses *run-length encoding* to efficiently store runs of black and white pixels in horizontal or vertical direction. This technique is used to compute horizontally and vertically smeared images that are obtain by thresholding runs of (white) background pixels. Combining these smeared images using a logical AND operation results in a coarse segmentation mask that can be refined further. However, it should be noted that the smearing process in the RLSA can also be considered a grouping process and therefore the method is sometimes categorized as a bottom-up approach.

X-Y cuts by Nagy et al. [NS84], also referred to as Recursive X-Y Cuts (RXYC) algorithm, uses projection profiles to iteratively split whole document pages in horizontal or vertical direction until no further splits are possible. Like RLSA the RXYC algorithm requires the input images to be binary and deskewed. Furthermore, page segmentation approaches that are based on projection profiles, the RLSA or other filtering approaches make assumptions about the document layout. As stated by Eskenazi et al. [EGKO17] this kind of algorithms are only able to deal with predefined classes of layouts (e. g. Manhattan).

Another group of top-down methods utilizes background information for segmenting document pages. Pavlidis and Zhou [PZ91] propose a white stream-based segmentation, and Baird [Bai92] uses maximal white rectangles to isolate text regions. However,

Figure 2.4: Representation of the area Voronoi diagram, courtesy by [Kis14].

estimating the maximal white rectangles that cover the background of a document is a complex task [Kis14]. Therefore, Breuel [Bre02] proposed an efficient geometric algorithm using a *branch-and-bound* methodology that is simple to implement.

The concept of using background analysis is further improved by Antonacopoulos [Ant98] by proposing a method using tiles instead of rectangles. The method uses smearing to coarsely locate regions of text and other foreground objects. Then white runs are concatenated in vertical direction to form white tiles that cover the background. Finally, the page segmentation is done by connecting the contours of the tiles and extracting the text regions surrounded by them. This method is capable of dealing with non-Manhattan layouts and is robust to skewed text lines.

A method that is similar to the white tiles approach is proposed by Kise et al [KSI98]. The approach uses an area Voronoi diagram to represent the structure of the background. Figure 2.4 shows an example of an area Voronoi diagram for a simple text image. In order to segment the document page edges of the diagram are removed based on simple statistics derived from a CC analysis. As an advantage of using the area Voronoi diagram this method is rotation invariant.

The previous brief summary of the top-down methods shows that this processing strategy usually relies on knowledge about the actual document structure. Ramel et al. [RDB07] therefore refer to these methods as model driven methods. Using *a priori* knowledge means that the methods are limited to certain types of document layouts or a complex architecture is developed to deal with varying layouts (e.g. the syntactic system for technical journals by Nagy et al. [KNSV93]). However, the majority of state-of-the-art methods evaluated at the ICDAR page segmentation competitions [ACPP13a, ACPP15, CAP17] also utilize top-down information but in combination with bottom-up grouping processes. These approaches are discussed in the section about hybrid page segmentation.

2.2.2 Bottom-up Methods

Bottom-up methods use primitives that are grouped together to form higher level structures of the page. Kise [Kis14] et al. classifies the RLSA and the Voronoi diagram based approach [KSI98] for page segmentation as top-down methods. The reason for this is that both algorithms subdivide document pages into intermediate regions (text lines/blocks, figure, tables, etc.) that might be split further depending on their type. However, the extraction those regions (smearing, Voronoi edge removal) can also be seen

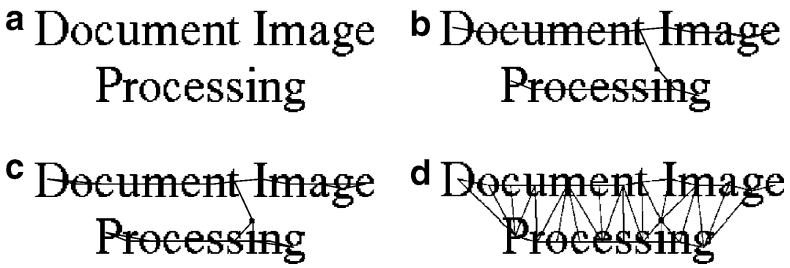


Figure 2.5: Representations of adjacency relations among CCs. (a) Original image, (b) Minimum Spanning Tree (MST), (c) k-NN (k =2), and (d) Delaunay triangulation, courtesy by [Kis14].

as a grouping process. Therefore, RLSA and the Voronoi based approach are sometimes also classified as bottom-up methods.

In binarized document pages CC can be used as primitives for the grouping process of bottom-up methods. O'Gorman [O'G93] proposed an approach called the document spectrum, also called *Docstrum*. This method connects the centroids of CC to each other using a k-Nearest Neighbours (k-NN) clustering based on Euclidean distance resulting a graph representation of the text regions. Figure 2.5c) shows an example of the representation of the adjacency relation using k-NN. The parameter k for the k-NN clustering is set to five and therefore typically includes edges between characters within text lines and between neighbouring text lines. The information of the adjacency relations is then gather in a 2D plot, the *Docstrum*, that visualizes the distance and orientation of all the edge in computed graph. Based on the Docstrum information about character spacing, text line orientation and text line spacing are derived and used to segment the document page.

The method by Simon et al. [SPJ97] is designed to be efficient and to able to deal with complicated layouts. It improves on the Docstrum method by introducing a new distance function for CCs and speeding up the computation. The distance function is based on the maximum distance between the edges of the Bounding Box (BB)s of the CC. Then the document structure is deduced by building a MST where the vertices represent CC and the edges between them are weighted according to the distance function. Figure 2.5b) shows an example of the representation of the adjacency relation using a MST. Different levels of the layout (i.e. words, line, paragraphs) are found in an efficient way by using heuristics and adaptive thresholding in the segmentation process.

An alternative to using k-NN, or MST in the CC clustering process is to use the *Delaunay* triangulation. This triangulation can be constructed as the dual graph of the area Voronoi diagram, as described by Kise [Kis14]. The advantage of this representation is that CC are connected to their neighbours in every direction. Figure 2.5d) shows an example of the representation of the adjacency relations using the Delaunay triangulation. This means that the Delaunay triangulation contains information about the local neighbourhood which can be used to segment the document image by deleting edges. An example for a

segmentation approach based on Delaunay triangulation is proposed by Xiao and Yan [XY03]. In this approach edges are statistically analysed and removed based on their length and orientation.

Other than using CC as primitives, methods based on texture analysis use individual pixels or patches of an image for feature extraction. Then based on the extracted texture features image regions are classified in different types of regions and grouped accordingly by using post-processing. Jain and Bhattacharjee [JB92] propose to use multichannel Gabor filters for classification of pixels into text or non-text. This concept is further improved by Jain and Zhong [JZ96] who propose to use feed-forward Neural Networks (NN)s to compute optimized filters for feature extraction. The learned filters are then used to distinguish between pixels representing images, background, text and line-drawings. Final segmentation results are then obtained by using morphological operations and computation of BB of the estimated regions. However, as stated by Kise [Kis14] texture analysis based methods tend to produce noisy results. Therefore, additional post-processing is required to compute final segmentation regions but on the other hand texture analysis can be applied to documents with overlapping layouts.

The advantage of using bottom-up approaches is their flexibility. They do not rely on a priori knowledge about the document layout but use local information to form higher level structures from primitives. Therefore Ramel et al. [RDB07] refer to these methods as *data-driven* methods. As discussed in this section bottom-up methods can be used to process documents with non-Manhattan and overlapping layouts. On the other hand bottom-up approaches are affected by problems such as high number of parameters, sensitivity to image noise and high computational complexity.

2.2.3 Hybrid Methods

Hybrid page segmentation methods combine techniques from bottom-up and top-down processing methods. The goal of hybrid approaches is to utilize the strengths of both strategies and provide improved segmentation performance. As indicated by the organizers of the ICDAR page segmentation competitions in 2013 [ACPP13a, ACPP13b] the participating methods share a similar methodology. The methods are based on bottom-up grouping processes but also use background and separator regions to convey the document layout information in a top-down way.

Among the best performing methods of the ICDAR page segmentation competition series is the so called *Fraunhofer Newspaper Segmenter* from Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS) at Sankt Augustin, Germany. The method won the competition in 2009 and participated in the following editions of the competition with slightly adapted versions of the original approach. It is based on a bottom-up grouping process of CCs which is guided by the *logical column layout*. The *logical column layout* is determined by computing vertical, and horizontal black separators and maximal white rectangles [Bre02] covering the background. Separation of text and non-text regions is achieved by using statistical analysis of the CCs. For the ICDAR page segmentation

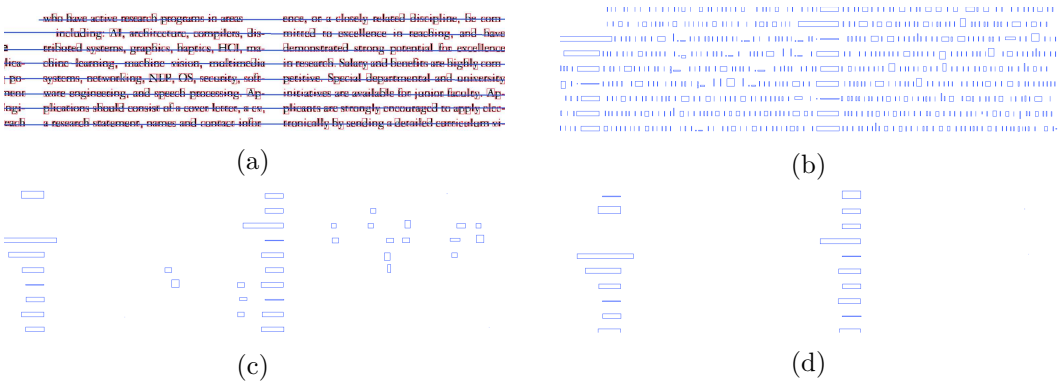


Figure 2.6: Illustration of the segmentation process of the method by Chen et al. [CYL13]. a) CC chains, b) white space rectangles between horizontally adjacent CCs, c) vertical runs of white spaces, d) refined set of white space runs. Illustrations courtesy by [CYL13].

competitions in 2011 and 2013 that are focused on historical document the preprocessing of the Frauenhofer segmenter was improved in terms of document binarization.

The method by Chen et al. [CYL13] achieved top results in the HBR2013 [ACPP13a]. Figure 2.6 shows an overview of the steps of the segmentation process of the algorithm. After binarization and the extraction of CCs white spaces between vertically adjacent CCs are extracted. Then runs of white space rectangles between vertically adjacent CCs are used to determine chains of CCs (see Figure 2.6a). Next white spaces between horizontally adjacent CCs are extracted (see Figure 2.6b). These white spaces are again grouped into runs (see Figure 2.6c) but this time in vertical direction. Then based on a statistical analysis final vertical runs are identified (see Figure 2.6d) which are used to cut CC chains into text lines.

Another example of a hybrid page segmentation approach is proposed by Ray Smith [Smi09]. This method is the basis for the layout analysis component of the open source OCR engine *Tesseract*. In the ICDAR page segmentation competitions the results of this method are used for comparison. The basic idea is to find tab stops and use them to determine the column layout of a document page. Then the column layout is used to group CCs with similar characteristics to form the final segmentation regions of the document.

In the ICDAR Competition on Recognition of Documents with Complex Layouts (RDCL) in 2015 and 2017 [ACPP15, CAP17] the best performing method with a success rate of 90.5% and 92.32% respectively is proposed by Tran et al. [TON+17]. They used a hybrid page segmentation technique that outperforms the other methods especially due to fewer (partial) miss errors. After binarization and extraction of CCs a classification of text and non-text is performed. This classification is based on the Minimum Homogeneity Algorithm (MHA) [TNK16] and further improved in 2017 [TON+17]. The algorithm performs classification in an iterative process that utilizes multilevel and multi-layer

homogeneous regions in combination with white space analysis. Then within the identified text regions text lines are extracted by grouping CCs based on white space analysis and paragraphs are identified in a final refinement stage.

2.3 Optical Font Recognition

The task of OFR is to identify the attributes that define the font used for representing text in a given document image. Several attributes like the typeface, weight, slope, size and others are used to specify a font. The use of different fonts in a document can have various reasons such as design choices, readability, structuring or others. However, the visual variations of different fonts can pose additional challenges for tasks such as OCR. Therefore, OFR can be used to address these challenges but also for gathering additional information which can be used for document indexing, information retrieval and other applications as stated by Pal and Dash [PD14]. Another OFR application is the extraction of logical layout information. In this application scenario text regions within a document image are distinguished according to their font. The recognition system proposed in this thesis uses OFR for logical labelling of text regions in order to identify regions representing different categories of information in a document image.

In this section a short overview of methods for OFR is presented with a focus on methods applicable for logical labelling of text regions. A more detailed overview of OFR is presented by the Slimane et al. [SKH+13] and in the work by Pal and Dash [PD14]. In the remaining part of this section OFR methods from two groups are presented. In Section 2.3.1 methods based on typographical features are discussed and Section 2.3.2 presents methods based on texture analysis.

2.3.1 Typographical Feature Approaches

Typographical feature are global attributes of text such as height, density and slant that are derived from text regions. There are different ways how these attributes can be extracted from document images. For example, Zramdini and Ingold [ZI93] proposed to use features derived from visual observations of the projection profiles of text lines. In Figure 2.7 the horizontal and vertical projection profile of a word image are shown. Based on these profiles five features are computed and used in combination with a Bayesian classifier in order to estimate font, weight, slope and size of text lines images. The presented experiments show that the method is able to accurately discriminate the font characteristics for varying font families with recognition rates between 88.5% and 100.0%. However, the evaluation of the method does not take into account the influence of the length of text samples and document degradation.

Zramdini and Ingold improved their system based typographical features in another work [ZI98]. They proposed to use eight features extracted from projection profiles and the CCs of a text line image. For the extraction of features from the CCs they are devised in different typographical classes according geometric properties relative to the text line.

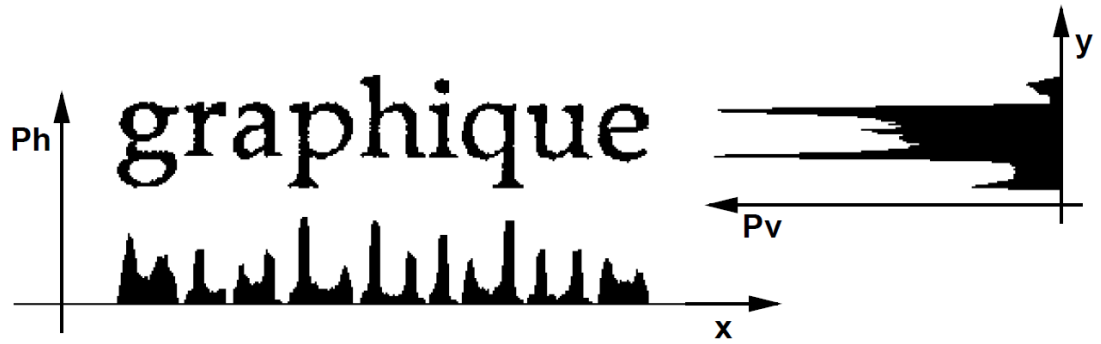


Figure 2.7: Horizontal (Ph) and vertical (Pv) projection profile of the word *graphique*, courtesy by [ZI93].

The classification is then done using a multivariate Bayesian classifier and trained using a set of 280 fonts (10 typefaces x 7 sizes x 4 styles). The method achieves recognition rates of 97% or higher for recognizing fonts and font attributes. Nevertheless, the authors also state that the training needs to be adapted in order to be able to deal with document degradation. Considering the length of text lines, they state that the recognition accuracy is decreased for shorter text lines but the method is applicable to lines with about ten characters.

An approach that computes style classification results on word level is presented by Ma and Doermann [MD04]. Under the assumption that OCR results are available they use six typographical features (stroke width, foreground density, etc.) extracted at character level. In the training step a features selection procedure is used to compute the optimal feature subset for each font style. Finally, the word classification is based on Gaussian Mixture Model (GMM) constructed for groups of characters with the same character code and a weighted majority vote over the characters of a word. The method achieves average recognition rates above 90% for bold and italic style detection. In addition, an evaluation with different levels of noise shows that the method is able to deal with degraded documents. On degraded images with an OCR accuracy of 87% or higher the bold and italic recognition rates are both above 90%. For images with lower quality many character strokes are broken which decreases the OCR accuracy as well as the font style detection. Therefore, the main drawback of the method is that it depends on the accuracy of the OCR.

2.3.2 Texture Feature Approaches

Mehri et al. [MGKH+13] provide an evaluation of texture features for the application of document segmentation. In their work they state that texture analysis can be used to extract region characteristic of an image and segment it into homogeneous regions. Therefore, texture analysis can be used for OFR. Furthermore, texture based segmentation approaches do not rely on *a priori* knowledge, such as structural (e.g.

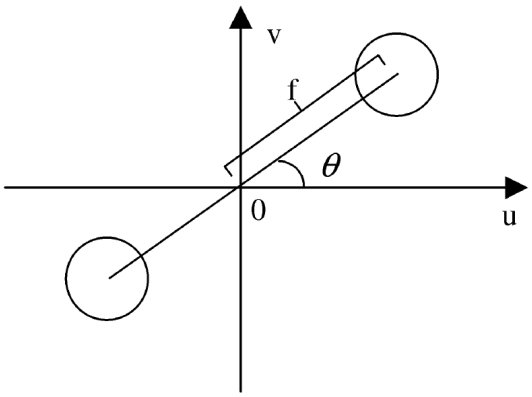


Figure 2.8: Illustration of the frequency response of an even-symmetric Gabor filter, courtesy by [ZTW01].

layout) or typographical (e.g. font size) information of the document.

There are several approaches to extract texture information from images such as the Grey Level Co-occurrence Matrix (GLCM) [HSD73], the autocorrelation function, Gabor filters [Tan92] and others. The autocorrelation function [MGKHM13] for example is used for the segmentation of entire books. However, Mehri et al. [MGKH+13] demonstrated in a comparative study (investigating GLCM, autocorrelation and Gabor filters) that Gabor features provide the best segmentation performance with a mean homogeneity accuracy of 95% on a data set with historical document images. In addition, the authors of the study also state that Gabor filters can be used to segment textual documents with distinct fonts.

Tan [Tan92] proposed to use a pair of Gabor filters to mimic the human visual system and extract texture information from images. Based on this work Zhu et al [ZTW01] designed a multichannel Gabor filtering approach that is used for font recognition in text blocks. Multichannel Gabor filters are created by varying the frequency f and orientation θ of the Gabor filters. As shown in Figure 2.8 the parameters define the appearance of the filter in the frequency domain and therefore can be used to cover different parts of the frequency plane. The authors do a comparison of their approach with the method based on typographical features presented by Zramdini and Ingold [ZI98]. The results show that with a performance of 94% (compared to 98%) the texture-based approach provides comparable results in terms of style classification. However, the method is outperformed in terms of the identification of typefaces with a performance of 83% (compared to 97%). It is concluded that the approach is able to identify global font attributes but has difficulties in distinguishing fine typographical differences of varying typefaces within the same font family. An advantage of the multichannel Gabor filtering approach is its robustness to noise.

Another approach based on Gabor filtering is presented by Ma and Doermann [MD03].

They use a multi-class classifier that is trained for a given set of fonts in order to provide word-level results. The evaluation of the approach shows that the method is able to classify different scripts and simultaneously classify a single script into multiple font styles. The method achieves a mean recognition rate of 83% for script as well as style classification. Factors that negatively influence the performance of the method are the insufficient amount of texture features in certain words and the incorrect segmentation of the word images.

2.4 Optical Character Recognition

In the recognition system proposed in this thesis OCR is used to recognize the text contained in the regions of interest that are identified in the layout analysis step. The task of OCR was one of the first application scenarios in DIA and influenced the development of this field of research, as described by Baird and Tombre [BT14]. The OCR methods and systems proposed in literature were first targeted at images of machine printed text. Through the advances in the field of machine printed OCR this task is seen as a solved problem under ideal conditions, as stated by Doermann and Tombre [DT14].

In the HBR2013 one evaluation scenario considered the performance of whole recognition pipeline (including text recognition) of the submitted methods for the recognition of historical books. As the organizers of the competition state OCR has been largely abandoned by the academic community. The reason why the text recognition abilities of the methods proposed in the HBR2013 are evaluated is that historic documents pose specific challenges that require additional improvements for common OCR methods.

However, only one of the proposed methods in the HBR2013 actually provided an OCR solution. The results of the challenge showed that state-of-the-art commercial (ABBYY FineReader®Engine 10 [Fre]) and open source (Tesseract 3 [tes]) OCR solutions outperformed the submitted academic method. The organizers concluded that the state-of-the-art methods are more flexible and differ in terms of preprocessing of the document images.

In the RDCL 2017 [CAP17] the best performing method [TON+17] in terms of segmentation performance adopted the OCR module of Tesseract [tes] in order to provide text recognition results. The evaluation results of the competition revealed that the improved segmentation results of the proposed method caused slightly improved OCR results when compared to using solely Tesseract with its built-in layout analysis module.

Based on the evaluation results in terms of OCR presented in the ICDAR page segmentation competitions [ACPP15, CAP17] the OCR module of Tesseract [tes] is adopted for the task of OCR.

CHAPTER 3

Methodology

The recognition system proposed in this thesis consists of three main components, which are page segmentation, font style classification and optical character recognition. An overview of the proposed system is shown in Figure 3.1. The page segmentation and font style classification components deal with the extraction of layout information. First page segmentation is used to extract text regions and derive structural information of the page by identifying text lines and blocks. Then font style classification is used to refine the segmentation of the text regions further. Text lines or parts of lines are labelled according to the logical document structure derived by analysing the font style of the text regions in combination with document (i.e. catalogue) specific knowledge. This means that the words contained in the text lines are assigned to specific categories of information such as the name of an artist or artwork based on their font style. In the last step the OCR component is used to extract the textual content of the segmented and labelled text regions. This component is based on the publicly available OCR module of Tesseract [tes].

The previously mentioned components of the recognition system build upon each other. In order to compute end-to-end recognition results the intermediate results of the individual stages and document specific knowledge are combined. The final labelling stage uses all the information gathered by the previous components of the recognition system. This means that the text contained in the segmented regions is labelled according to the font style information and document knowledge. The resulting labelled text regions represent data entries of different categories (e.g. ID, artist, address, etc.) contained in the catalogue document images. Finally, this extracted information is output such that it can be used for further processing.

In the following sections of this chapter the methodology used to develop the individual components of the recognition systems are described in detail. Section 3.1 details the page segmentation approach used to extract physical layout information. In Section 3.2 the methodology for font style classification is presented. Finally, Section 3.3 explains

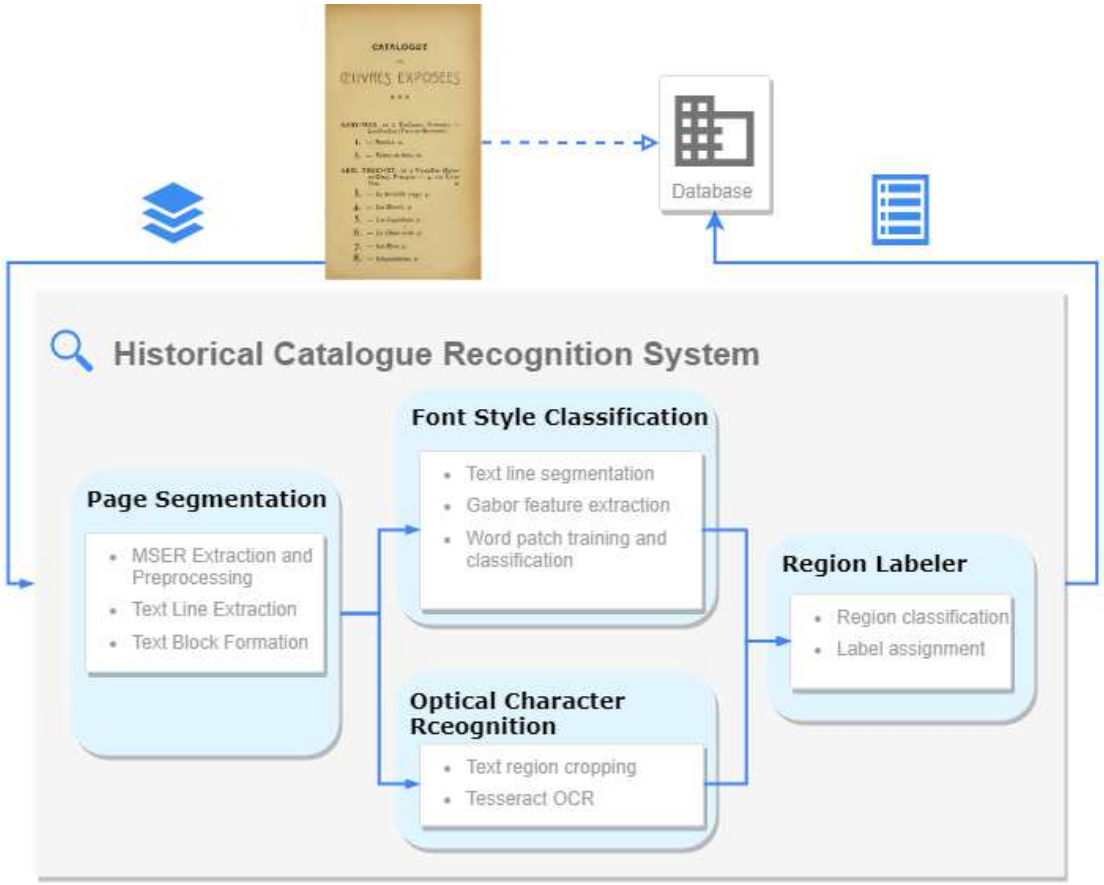


Figure 3.1: Overview of the recognition system showing the four components of the system - *page segmentation*, *font style classification*, *optical character recognition* and the *region labelling* - and how they are used in the workflow of the system.

the process of combining the gathered layout information with the OCR methodology of Tesseract in order to compute the final recognition results.

3.1 Page Segmentation

In this section the process of extracting geometrical layout information within the recognition system is described. As shown in Figure 3.1 the aim of this component is to provide a segmentation of a document image into text line regions that is used by the other components. For this purpose, candidate text regions are extracted and grouped to form text line regions.

In state-of-the-art pages segmentation methods [CYL13, TNK16, TON+17, Smi09] the extraction of text regions is based on a binarization of the input images and a CC analysis. In contrast to this methodology, region detection methods can also be used to extract

candidate text regions without additional binarization.

The proposed system uses the MSER algorithm [MCUP04] to extract regions of interest from input images. This approach is not used by other methods in the page ICDAR page segmentation competition series. However, MSER are used in the generic text line extraction method by Wang et al. [WFS+15] targeted at a wide variety of document images and in scene text applications [NM12]. Using MSER allows to extract text regions in images with varying background and at arbitrary orientations. Therefore, this method can also be used for the extraction of text regions in challenging historical document images.

After the extraction of the candidate text regions they are analysed and grouped into text lines. A hybrid methodology is adopted, since this class of methods had the best results in the ICDAR page segmentation competitions [ACPP13a, ACPP13b, ACPP15]. The proposed approach is based on a white space analysis which is used to guide the text line extraction and combined with an additional separator detection.

Following the extraction of text lines another grouping process is applied in order to form text blocks. Furthermore, post-processing is applied to in order to improve the segmentation results. Based on these results the proposed method is compared against state of the-art methods using the evaluation scheme of the HBR2013 [ACPP13a].

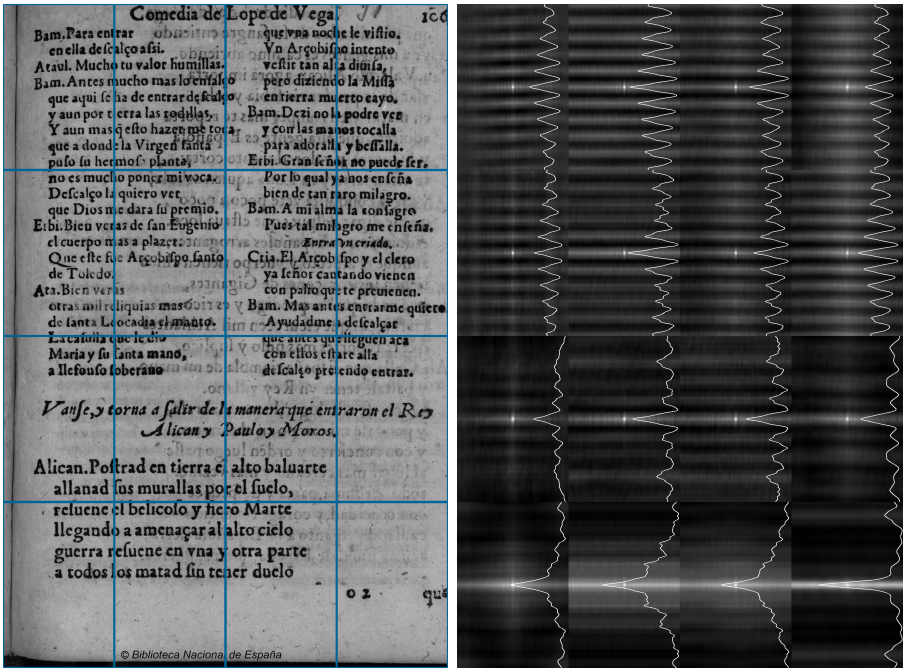
In the following sections the major stages of the proposed segmentation algorithm are explained in detail. In Section 3.1.1 the extraction of text regions and additional preprocessing is explained. Section 3.1.2 is concerned with the extraction of text lines. And finally, the text block formation and further post-processing is described in Section 3.1.3.

3.1.1 Text Region Extraction and Preprocessing

The first step of in the recognition system is the extraction of candidate text regions in a given input image. In order to be able to deal with historical documents containing varying backgrounds the proposed system extracts MSER in order to find candidate text regions. In contrast to other page segmentation methods the advantage of this approach is that no additional binarization is needed. On the other hand, an additional input image scaling and a pruning stage are used to reduce the amount of extracted candidate text regions.

Text Height Estimation and Input Image Scaling

The resolution of the input images, i.e. the size of the text regions, influences the number of extracted MSER. For higher resolutions single characters tend to be split into several candidate regions due to local variations. Therefore, the proposed system uses text height estimation to perform an adaptive scaling of the input images. This reduces the number of broken character regions and unifies the workflow for images with different resolutions.



(a) (b)

Figure 3.2: Illustration of the Athena text height estimation process: Figure (a) shows the subdivision of an input image and Figure (b) shows the extracted NACF results and their horizontal projection.

Based on the assumption that the processed document images have one predominant text height two methods for text height estimation are tested. The first approach is based on the Normalized Autocorrelation Function (NACF) and is called *Athena* [PYR13]. The method uses a multiscale representation that divides an image into an increasing number of subimages. For each of the subimages the NACF is computed and based on a horizontal projection the predominant spatial frequency is determined. The frequency information is then used to reassemble a Probability Mass Function (PMF) of the text height estimate for each scale. Finally, the text height estimate is determined by summing up the PMFs over all scales and finding the value with the highest probability.

In Figure 3.2 the extraction of the NACF and the spatial frequency information used by the Athena method is illustrated. An input image and its division into subimages at a fixed scale is indicated in Figure 3.2a. The corresponding results of the NACF for each subimage are indicated in Figure 3.2b. Additionally, within each subimage the horizontal projection profile used for determining the predominant spatial frequency is shown too.

As an alternative for the Athena method a rough estimate of the text height can be computed based on the extraction of MSER (see next section). The rough text height estimate is defined as:

$$rth = m_h \cdot 2 \tag{3.1}$$

where m_h is the median height over all extracted text regions. The equation is based on the assumption that the median region height roughly corresponds to the so called *x-height* of the predominant font, i.e. the height of lower-case letters excluding ascenders and descenders. This assumption has been verified experimentally on the images of the HBR2013 test data set. The ratio between the x-height and the overall font height varies for different fonts. For the rough text height estimation the text height is defined as double the x-height (median), see Equation 3.3. This rough estimate is determined by measuring the ratio on synthetic text images of 172 different font families. The average ratio computed over all fonts is 1.8 which is rounded to 2.0 in the final equation.

For the Athena method an additional check is used to verify the quality of the text height estimation. Applying this additional check is required in order to deal with cases where the computation of the Athena method yields erroneous results. The check is based on the standard deviation of the PMF that is used to compute the text height estimate. A high standard deviation indicates that the result of the text height estimation is likely to be wrong. If the threshold for the standard deviation is exceeded the rough text height estimation method is used as a fallback solution. Without the additional check the wrong text height estimation results would negatively influence the segmentation performance on the HBR2013 test data set due to excessive downscaling of certain images.

For both methods the input image is scaled so that the estimated text height corresponds to a predefined maximum height in the rescaled image. If the estimated text height is lower than the predefined maximum height the image is not scaled. In the proposed system the value of the maximum height parameter is set to a standard value of 50 pixels. This setting is able to reduces the amount of split character regions and is applicable to a wide range of documents as in the HBR2013 test data set. Adapting this parameter to a specific data set might improve the results further. In Figure 3.3 the estimated text height values for images from the HBR2013 training data set are shown. The images show that the results of the two methods are very similar. Text height values estimated by the Athena method are indicated using blue squares and the results of the rough method are shown in green.

Based on the evaluation results presented in Section 4.1 Table 4.2 the use of the two text height estimation methods result in an increased segmentation performance. The results of the two methods are rather similar but the rough text height estimation provides superior results. In terms of segmentation performance the rough text height estimation results are 0.5% better than the Athena results.

In addition, the *Athena* method uses the rough estimation method as a fallback solution on some images in the HBR2013 test data set. This means that both methods are required when using the Athena method and after the rescaling process the MSER need to be extracted. When using the rough estimation method only an additional MSER extraction pass is required in order to rescale the input image. Furthermore, if an image

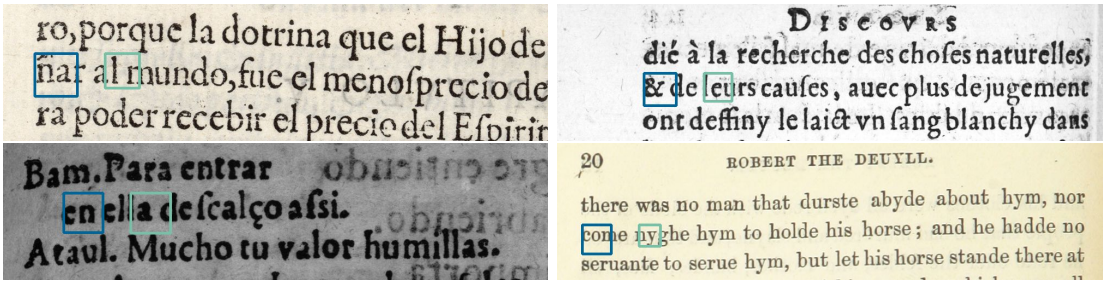


Figure 3.3: Images showing results of the text height estimation. Results of the Athena method are shown in blue, results of the rough method in green.

does not need to be rescaled according to the estimated text height no additional MSER extraction pass is required. In this case the extracted MSER are directly passed on to the further processing stages of the page segmentation workflow. Hence, the rough text height estimation process can be integrated into the page segmentation workflow effortlessly.

Due to the better segmentation performance the proposed system uses the rough text height estimation method as standard. For data sets with a fixed resolution (e.g. images from a single catalogue, with fixed image acquisition setup) the text height estimation can be omitted and a constant scale parameter is defined in advance.

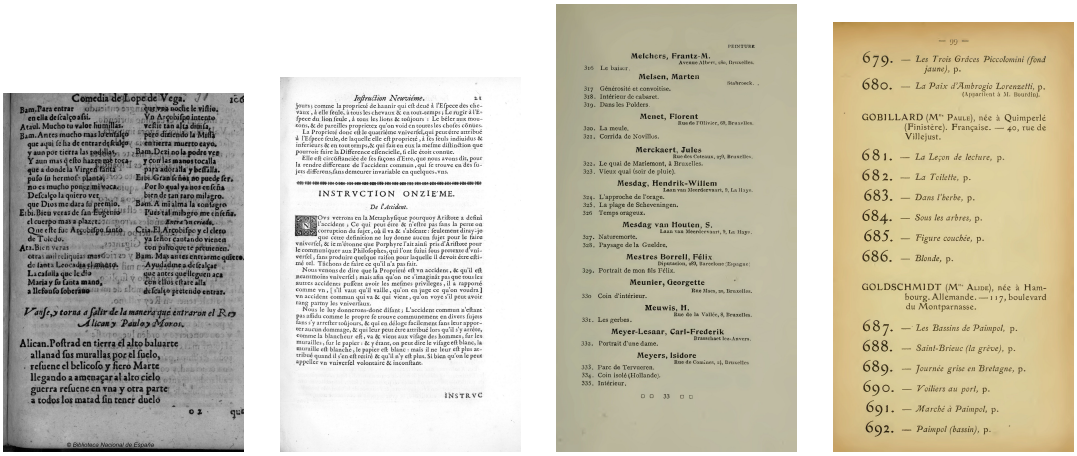
MSER Extraction

MSER is a robust region detector introduced by Matas et al. [MCUP04] originally designed for finding of stereo correspondences in image pairs. However, the method can also be used for text detection. The basic idea of the MSER algorithm is to use repeated thresholding operations to extract a set of CCs, so called Extremal Region (ER). In the proposed system the standard implementation based on the original work [MCUP04] by Matas et al. is used.

The parameter $\triangle$ determining the number of consecutive thresholding operations is set to the standard value 5. Adapting this value to the specific data set might improve the results but by default the parameter is not adapted to the input data. The only addition that is made to the original approach is an additional filtering of duplicate regions. The final result of the MSER extraction is a set of nested CCs as shown in the example in Figure 3.5a.

MSER Pruning

The MSER extraction process creates a larger number of candidate text regions compared to a CC analysis. This is due to the fact that a region can have multiple stable thresholds resulting in sets of nested CCs. In order to reduce the number of extracted candidate text regions a pruning of the regions is applied. The pruning is based on assumptions



(a) (b) (c) (d)

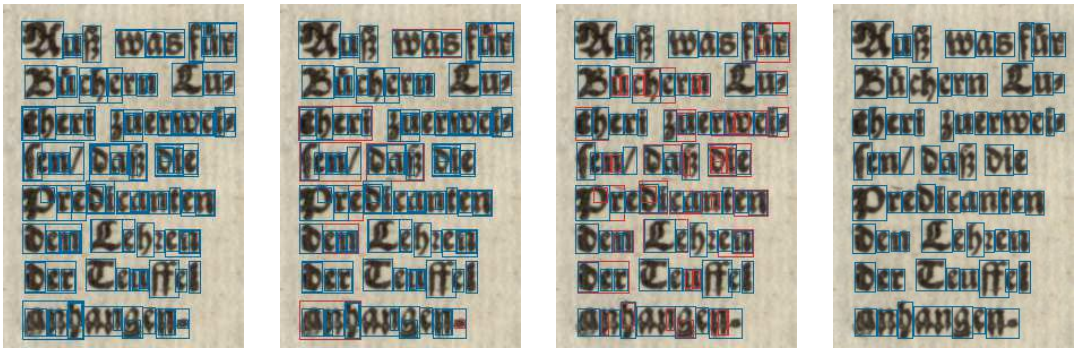
Figure 3.4: Images showing examples of input images from the HBR2013 training data set (a, b) and from historical catalogues (c, d), courtesy by [Arcc, Arcb].

about the input data and adapted for the purpose of layout analysis. The following assumptions about the input data are used for the pruning process:

- Input images represent single document pages
- The document pages have one predominant text height
- Text regions are represented by dark (black) regions
- Background regions are represented by bright (white) regions
- Lighting changes, such as specular highlights, do not affect the images

These assumptions are true for the images from the historical catalogues but can also be applied to the data used in the HBR2013 competition. As shown in Figure 3.4 the layout of the input images varies but the basic assumptions about the text regions are true for images from both data sets. The last assumption is based on the fact that the images are acquired under controlled settings for the purpose of digitization.

Based on the assumption that text is represented by dark foreground regions the second pass of the MSER algorithm, using the inverted input image, is omitted. This second pass is only needed to detect regions of negative text, which are neglected since they are considered to be graphical or background regions. Furthermore, based on the aforementioned assumptions, colour information from the input images is not utilized and MSER are extracted using a greyscale representation of the input images. In Figure 3.5 the text region extraction process is illustrated on an exemplary image patch. The initial text regions without additional pruning are shown are shown in Figure 3.5a.



(a) (b) (c) (d)

Figure 3.5: Images illustrating the pruning process of MSER for an image patch from the HBR2013 test data set. Image (a) shows the initial set of MSER, (b) shows regions pruned due to size constraints, (c) illustrates the removal of nested regions and (d) shows the final text regions after the pruning process.

The size of the extracted MSER is further restricted by a statistical analysis of the region properties. Based on the analysis of the size of the regions upper and lower bounds are computed and used for filtering. The bounds are computed using the following equations:

$$lb = (Q_2 - (Q_3 - Q_1)) \cdot \frac{2}{3} \tag{3.2}$$

$$ub = (Q_2 + (Q_3 - Q_1)) \cdot 2 \tag{3.3}$$

where Q_1, Q_2, Q_3 denote the quartiles of the distribution of the heights or widths of the candidate text regions. Based on the above equation lower bounds h_{lb}, w_{lb} and upper bounds h_{ub}, w_{ub} are computed. The extracted MSER are then pruned according to these bounds, meaning that width and height of the BBs must be within the respective bounds. In Figure 3.5b the regions pruned due to these size constraints are shown in red.

Considering the page segmentation procedure that is used to extract lines and blocks of text another pruning step is used to remove nested regions. The text region grouping is discussed in detail in Section 3.1.2. However, a main aspect of this process is the extraction of white spaces. In order to ease the white space extraction areas with nested or largely overlapping MSER are resolved.

This is done by applying a pruning scheme that preserves foreground areas but also splits regions if they can be composed by smaller regions. First all regions are sorted according to the size of their BB from biggest to smallest. Then each BB is compared with all others that are smaller and regions that are overlapped by the current one by more than 75% are marked for further processing. The threshold of 75% makes sure that regions having one or more common outer boundaries are also detected. Finally, a region is

removed if the union of the BBs of its overlapping regions covers more than 75% of the pixels BB. If the condition is not fulfilled all overlapping regions are removed and only the region under investigation is preserved. The pruning of nested regions is illustrated with the regions shown in red in Figure 3.5c.

The text regions remaining after these pruning steps represent the final set of text regions that is used in the following processing step of the page segmentation procedure. The remaining text regions for the exemplary image patch are shown in Figure 3.5d. The example shows the reduction of the amount of text regions but the remaining regions are still sufficient to cover the textual areas of the image patch. This means that the pruning results in a more efficient representation of the text areas.

3.1.2 Text Line Extraction

The MSER extracted in the previous stage are grouped in order to form text lines and deduce further layout information. This process is inspired by the page segmentation method by Chen et al [CYL13]. It is based on the extraction of an initial set of candidate text lines, a white space analysis and an additional black separator detection.

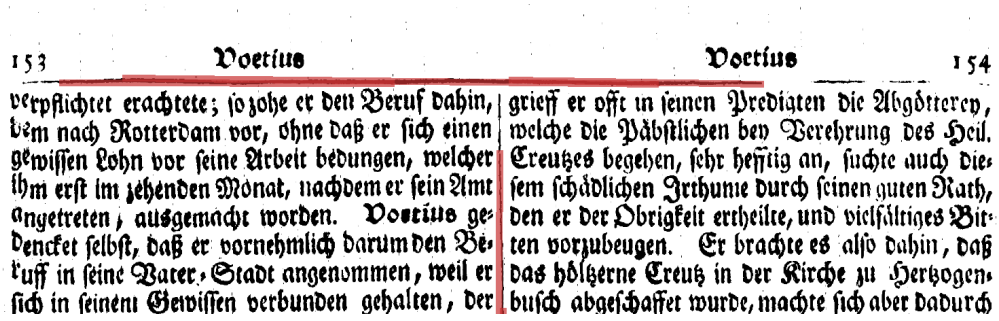
Black Separator Detection

Before the text regions are grouped into text lines so called black separators are detected. In the context of the proposed system black separators are dark, elongated horizontal, or vertical lines that are used to constrain the layout of the document page. Examples are horizontal lines used to separate columns of text or vertical lines delimiting header regions.

Black separators are detected using the Line Segment Detector (LSD) method [VGJMR10]. The detected lines are filtered with respect to the estimated median text region height. Only separators with a length bigger than four times the median text region height are considered in the page segmentation process. According to Equation 3.3 this length corresponds approximately to two times the predominant text line height. This threshold is used to avoid detecting short black lines caused by headings, touching text line regions or graphical regions that do not represent separator regions. The remaining filtered separator lines are saved in a list and used to constrain the segmentation process of the following stages.

An example showing black separators extracted from a document image is shown in Figure 3.6. The semi-transparent red lines indicate the estimated separators. As shown in the image some parts of separators are missed because the black separator lines are interrupted due to bad image quality.

In the text line extraction process all connections that cross a black separator are neglected. The same is true for all other steps that merge text regions. This means, although the use of black separator information is not explicitly mentioned it is always utilized and contributes to overall performance of the page segmentation. Applying the


153 Voetius Voetius 154

verpflichtet erachtete; so zohe er den Beruf dahin, dem nach Rotterdam vor, ohne daß er sich einen gewissen Lohn vor seine Arbeit bedungen, welcher ihm erst im zehenden Monat, nachdem er sein Amt angetreten, ausgemacht worden. Voetius gedencket selbst, daß er vornehmlich darum den Beruff in seine Vater-Stadt angenommen, weil er sich in seinem Gewissen verbunden gehalten, der

griff er offt in seinen Predigten die Abgötterey, welche die Päbstlichen bey Verehrung des Heil. Creutzes begehen, sehr heftig an, suchte auch diesem schädlichen Irthume durch seinen guten Rath, den er der Obrigkeit ertheilte, und vielfältiges Bitten vorzubeugen. Er brachte es also dahin, daß das höltzerne Creutz in der Kirche zu Hertzogenbusch abgeschaffet wurde, machte sich aber dadurch

Figure 3.6: Results of black separator detection. Separator lines are shown in semi-transparent red colour.

black separator detection improves the segmentation performance by 0.58% as shown in Table 4.4 in Section 4.1. These results show the influence of using black separators to split the initial text line hypotheses without using an additional white space segmentation step. The majority of the resulting splits can also be detected by the white space segmentation step described later in this section. However, the black separator detection is used as an additional step that increases the robustness of the system.

Text Region Grouping

The first step of the text line extraction process is to group text regions into horizontal chains. This is based on the assumptions that text lines are aligned horizontally and the text regions forming them are vertically overlapping. In the hybrid method proposed by Chen et al. [CYL13] chains of text regions are built by linking each text region with its Right Nearest Neighbour (NN_r). In the proposed system this concept is adopted and further extended.

Instead of finding only one NN_r multiple NN_rs are extracted and used for clustering the extracted MSER into text lines. For the computation of theses NN_rs the BBs of the extracted text regions are compared to each other by the following procedure. The BBs are sorted in ascending order according to the y-coordinate of their top. Then each BB is compared to the others until a BB with a bigger index is reached that has no vertical overlap with the current BB. The subsequent boxes can be skipped as their minimum y coordinates are even higher. All other boxes that have a vertical overlap are considered as neighbours. In addition, further conditions for the final linking of neighbouring text regions are employed:

$$dist_x(t_1, t_2) < m_h \cdot 5 \tag{3.4}$$

$$max(h_{t_1}, h_{t_2})/min(h_{t_1}, h_{t_2}) < 3 \tag{3.5}$$

$$overlap_v(t_1, t_2)/max(h_{t_1}, h_{t_2}) > 0.3 \tag{3.6}$$

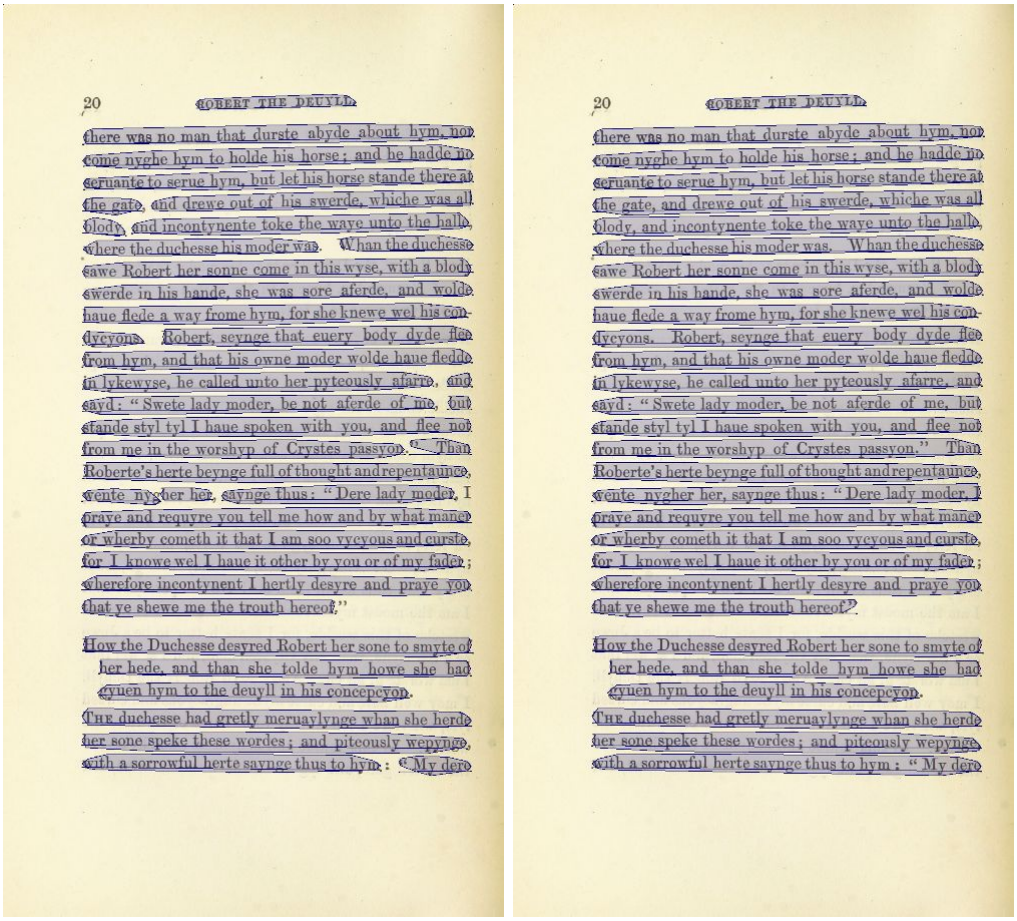
The first condition (3.4) limits the maximum vertical distance $dist_x$ between two neighbouring text regions t_1, t_2 to five times the median text region height m_h of the document. The second condition (3.5) makes sure that the ratio of the heights h_{t_1}, h_{t_2} of two neighbours is not bigger than 3. The third condition (3.6) finally makes sure that the vertical overlap is bigger than 30% of the height of the taller component. These conditions are based on assumptions about the document layout:

The threshold for the maximum vertical distance is set such that text lines containing large gaps caused by irregular spacing, missed character regions or hyphens are not split into multiple line fragments. Text regions that are linked across gaps between columns or other separators are split later on in the segmentation process. The condition for the ratio of the heights is based on the knowledge about the relation between the x-height and the text line height. Based on the Equation 3.3 the maximum height ratio has to be higher than two in order to be able to group characters along a text line. Therefore, the maximum for the height ratio is set to three to deal with a wide range of fonts and inaccuracies in the region detection process. Finally, the third condition is also based on the second condition and the assumption that text lines are aligned horizontally. Two horizontally aligned text region with a max height ratio of three should have a vertical overlap of at least one third. Therefore, the minimum threshold is set to 0.3 which again tolerates minor inaccuracies. Applying this condition enforces links between text regions by exploiting basic layout assumptions and additionally reduces erroneous links with noise or background regions.

In order to support the grouping process for each text region a maximum of three NN_rs is computed and saved in a graph representation. If more than three NN_rs are found only the nearest three neighbours according to the Euclidean distance between the region centres are used. The advantage of computing multiple NN_rs is that the grouping process of text regions can be designed in a robust way. With multiple links it is possible to deal with outliers, noise and other situations that might cause errors if only one neighbouring text region is considered for linking.

In Figure 3.7 the results of the text region grouping process on an image from the HBR2013 training data set are shown. Figure 3.7(a) shows the results for using one NN_r and Figure 3.7(b) for three NN_rs. The comparison of the results shows that the use of three NN_rs is better at clustering text regions into text lines. The use of one NN_r on the other hand results in an oversegmentation of the text lines.

In Figure 3.8 two examples of text regions and their corresponding NN_rs are illustrated. The NN_rs are indicated with red lines connecting the centres of the linked text regions. Both examples in the Figure show that the nearest linked text region is small and not aligned with the centre line of the corresponding text line. In the example in Figure 3.8a the linking process of the text regions is not affected by the small text region, i.e. the comma, because the small region is linked with the characters of the following word. In



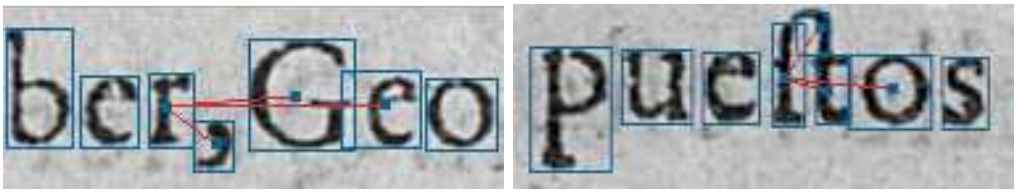
(a) (b)

Figure 3.7: Images illustrating the difference in the extracted text line candidates when using one (a) and three (b) NN_r in the text region grouping stage.

the example in Figure 3.8b the nearest linked text region is a part of a character that is erroneously split off. Due to the position of this split off region it is not linked with the following characters. In this case the linking process is continued using another NN_r avoiding an erroneous split of the text line.

For the clustering of text lines the computed links between text regions are first sorted according to their length in ascending order. Then the clustering process is started at the shortest link and simple heuristic rules are applied to extract initial text line hypotheses. When processing a link between two text regions three different cases are differentiated. Both text regions are unassigned, one text region is part of a text line but the other is not, or both text regions are contained in a text line.

Two text regions that are not clustered in a text line are merged together to form a new text line if their relative vertical overlap, as specified in condition 3.6, is bigger than 0.4.



(a) (b)

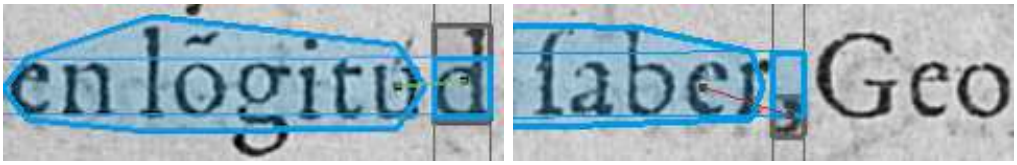
Figure 3.8: Two examples of text regions and their corresponding NN_rs computed in the text region grouping stage.

This means that in addition to the previous conditions for the adjacency computation an increased threshold value is used to form a text line cluster. The increased threshold value ensures that the text regions have a similar height and are likely to be aligned horizontally. Which is especially important for the first couple of regions since they are the basis for the text line grouping process.

In case of a link between two regions where one is contained in a text line a computation is used that validates whether the unassigned region should be merged with the existing text line. For short text lines that contain less than ten text regions components the validation computation is the same as for two unassigned text regions. Which means the validation is based only on the two linked text region components.

For text lines with at least ten components the validation computation additionally takes the properties of the text line into account. A method similar to the merging process of the *Text Spotter* method [QM16] is used. The difference is that the grouping process is started with a set of components that represent a text line candidate rather than with a single component. Therefore, an approximated centre line and the average component height are computed based on the current text line. Then like in *Text Spotter* a strip having the average text height is aligned with the centre line. This strip is finally intersected with the horizontal span of the BB of the unassigned text region. For the validation decision the Jaccard index between the region's BB and the intersection rectangle is computed. Where the Jaccard index is computed as intersection over union of the rectangle areas. If the Jaccard index is bigger than 0.5 merging the text region is considered valid. An additional distance threshold is not applied because the processed text regions need to fulfil the conditions 3.4, 3.5 and 3.6 anyway.

Figure 3.9 illustrates the validation process and shows the computation of the Jaccard index for two exemplary text regions. In the images the large blue polygonal region indicates the existing text line cluster and the grey rectangle indicates the BB of the unassigned text region. In addition, the narrow blue lines indicate the strip approximating the existing text line and the blue rectangle the intersection with the horizontal span of the text region. This means that the Jaccard index is computed using the blue and the grey rectangle. The validation process of the example region shown in Figure 3.9a yields a positive result. Whereas the example in Figure 3.9b yields a negative result because of



(a) (b)

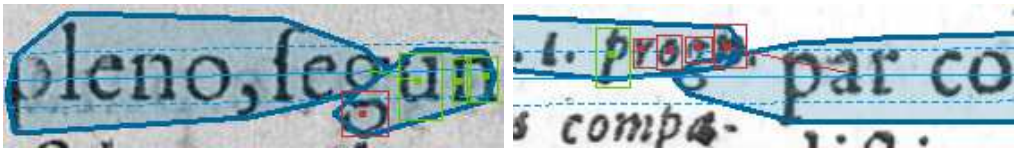
Figure 3.9: Images showing the validation process for merging two text regions if one region is already associated with a text line cluster. Figure (a) shows a valid merge, whereas Figure (b) depicts an unsuccessful merging operation.

the small area of intersection between the two computed rectangular areas.

Links between text regions that are both contained in text lines are validated similarly to merging an unassigned text region with a text line. However, short text lines are processed in another way. In case of both text lines containing less than ten, or one containing less than six text region components the validation process is the same as for two unassigned text regions. Otherwise the text line containing more text regions is selected and again a centre line and the average text region height is computed. Then the centre line is extended and the distance of up to five of the nearest text regions, contained in the other text line, is measured. Where the distance is the Euclidean distance of the text region centres to the extended centre line. Merging the two linked text lines is considered valid if the distance of the majority of the text regions is lower than half the average text region height. Considering multiple text regions in the linking process ensures that the transition between the text line clusters is continuous but still provides enough flexibility to deal with warped text lines (e.g. caused by page curl).

In Figure 3.10 two examples of the validation process for merging text line clusters are shown. The existing text line clusters are illustrated using blue polygonal regions and the blue dashed lines indicate the maximum acceptable distance from the centre line. Text regions within the acceptable distance are coloured in green and text regions invalidating the merging process are coloured in red. The example in Figure 3.10a shows a valid merge between two text line clusters, although one of the investigated text regions is considered too far away. In contrast to this the merging operation illustrated in Figure 3.10b results in a negative response because the majority of the investigated text regions vote for invalid.

After all the links between the text regions have been processed the resulting text region clusters represent the initial text line hypotheses used for further processing. In order to improve the text line hypotheses post-processing is applied. First unstable text lines are detected. A text line is considered unstable if the average orientation of text regions within the text lines differs from the orthogonal of the baseline for more than four degrees. Then an expanded ellipse is fitted to the unstable text lines and used to find overlapping stable text lines they should be merged with. In addition, a text line candidate is merged with another, bigger text line if more than 75% of its area is covered. This process is



(a) (b)

Figure 3.10: Illustration showing the validation process for merging two text regions where both regions are associated with a text line cluster. Figure (a) shows a valid merge and Figure (b) shows an example of an invalid merging operation.

used to merge small erroneous text line candidates located near text lines that are caused by diacritical symbols, split off text regions or punctuation marks. Finally, text lines containing less than three components are also removed.

Compared to the original approach by Chen et al. [CYL13] the text region grouping process is adapted in order to be able to deal with MSER used as candidate text regions. This is achieved by increasing the number of NN_rs and using a graph representation in combination with a set of heuristic rules. The benefit of these adaptations is that no additional binarization is needed and the grouping process is robust to outliers and noise.

The results of this initial text region grouping stage can be seen as the basic segmentation of the document image. Therefore, the results for the extracted regions represents the baseline of the segmentation performance. As shown in Table 4.4 in Section 4.1 the text region grouping stage without additional processing results in a basic segmentation performance of 56.44% on the HBR2013 test data set. The reason for this low value is the fact that the expected output results are blocks of text rather than single lines. Therefore, the text line results represent an oversegmentation which can be improved by grouping lines into text blocks. Applying the text block formation procedure detailed later in this section results in a segmentation performance of 69.64% for the basic text line regions. Comparing this result to the values in Table 4.1 in Section 4.1 shows that even the basic text line segmentation (including additional text block formation) is able to outperform the segmentation results of Tesseract 3 by 3.94%.

White Space Segmentation

The text line hypotheses that are extracted as described in the previous section can be segmented further based on a white space analysis. For this purpose, the white space based segmentation procedure by Chen et al. [CYL13] is adopted and adapted according to the specific requirements of the proposed system. The goal of this white space segmentation method is to identify white spaces that represent gaps between columns of text and use them to improve the segmentation results.

The first step of the white space segmentation process is the extraction of white spaces. For every text line white space rectangles filling the blank areas between the components of the text lines are extracted. In the next step a rough filtering is performed that reduces

the number of white spaces per text line. Based on the assumption that a document page has a maximum of eight columns per page the number of between-column rectangles within a text line is limited to N_{bcr}. N_{bcr} is defined as:

$$N_{bcr} = max(1, \frac{8 \cdot w_l}{w_p}) \tag{3.7}$$

where w_l is the width of the text line and w_p is the width of the page. Based on this formula only the N_{bcr} widest white spaces of each text line candidate are preserved. Then the remaining white space rectangles are used to compute an over-segmentation of the initial text line set.

For the remaining text line candidates and white space rectangles adjacency relations are computed. Isolated white spaces that are surrounded by text line regions in all directions are removed. In addition, white spaces rectangles surrounded by short text line regions are also removed if they are shorter than twice the median text region height. A text line candidate is considered as short if it contains less than five components or is shorter than five times the median text region height. This definition is based on the assumption that a short line contains only one word and an average word has five characters. The reduction of white spaces near short text line regions avoids over-segmentation and the additional upper bound ensures that no major white space gaps are removed.

The final stage of the white space segmentation is based on the identification of vertical runs of white space rectangles. This process is based on the adjacency relations. White spaces that are horizontally overlapping are grouped into white space runs and analysed in detail. For this purpose, the text line regions that are horizontally adjacent to the remaining white spaces are selected and their maximum gaps are computed. The maximum gap of a text line is the width of the widest white space rectangle that it contains.

The analysis of the white space runs proceeds as follows. Each white space run is compared with the maximum gaps of their neighbouring text lines. If the highest maximum gap value increased by 30% is higher than the width of a white space the corresponding white space is marked as a within-column candidate. Then white spaces runs that contain only within-column candidates are removed. In addition, leading or trailing white spaces of remaining white space runs that are marked as within-column candidates are also removed. According to Chen et al. [CYL13] these white spaces are likely to be *real* within-column candidates adjacent to column gaps. After the white space removal process the remaining text lines are merged accordingly and their maximum gaps are updated. Then the analysis of the white space runs is repeated until it converges and no further white spaces are removed.

As a final result of the white space segmentation the text lines split by the remaining white space runs are extracted. Text lines containing less than three components are removed. These regions are considered as noise that is split off during the white space segmentation process. The main difference of the proposed segmentation approach compared to the

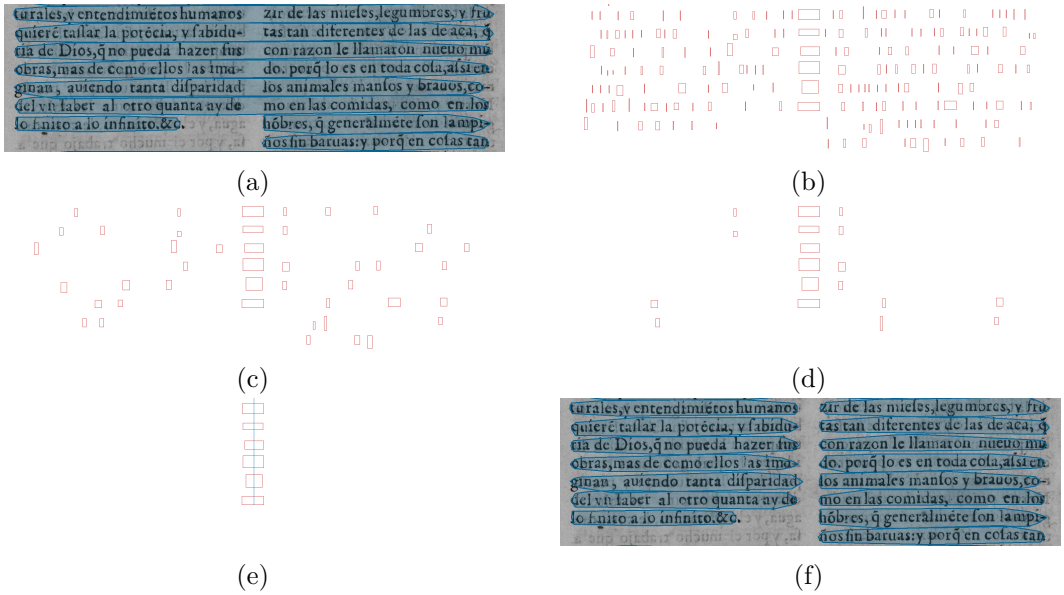


Figure 3.11: Illustrations showing the workflow of the white space segmentation process inspired by Chen et al. [CYL13]: (a) initial text lines, (b) white space rectangles of initial text lines, (c) white spaces after limiting maximum amount per line, (d) vertical runs of white spaces, (e) refined set of white space runs, (f) final set of text lines.

original work by Chen et al. [CYL13] is the use of MSER as initial candidate text regions. Due to the larger number of regions and corresponding inaccuracies in the white space extraction caused by using MSER minor adaptations are made. The first adaptation is that the maximum gap parameter is increased by 30% for identifying within-column candidates. In addition, in the original approach white spaces surrounded by text lines of one word are labelled as within-column candidates. This process is refined in the way that short lines (as defined earlier) are used instead of lines of one word because the identification of the latter type proofed to be inaccurate.

The individual steps of the white space segmentation process are illustrated in Figure 3.11. Figure 3.11a shows the initial set of text lines indicated with blue polygonal regions. In Figure 3.11b all white spaces contained in the initial text lines are visualized with the background removed for better visibility. The reduction of the amount of white spaces according to the parameter N_{bcr} is shown in Figure 3.11c. The elimination of isolated white spaces and extraction of white space runs is shown in Figure 3.11d and finally the result after the refinement of white space runs is shown in 3.11e. Based on the white space analysis results illustrated in the previous figures the final set of text lines is shown in Figure 3.11f.

By applying the white space segmentation procedure on the extracted text line candidates the segmentation performance of the proposed system is further improved. The refined text line segmentation increases the segmentation performance by 2.27% compared to

the basic text line extraction results. The comparison of the results is shown in Section 4.1 in Table 4.4.

Segmentation of Marginalia

The white space segmentation presented in the previous section is used to segment text regions into columns. This process takes into account local information and therefore represents a flexible way to identify white space regions. However, based on the basic assumptions of the method it is only able to detect gaps that are wider than the white spaces within the neighbouring text lines. Minor gaps that separate so called *marginalia* are not detected in the white space segmentation process. Marginalia are additional notes, comments and other text or non-text elements that are located in the margin of document pages. This kind of regions are contained in various types of historical documents and therefore they are also contained in the HBR2013 test and training data set.

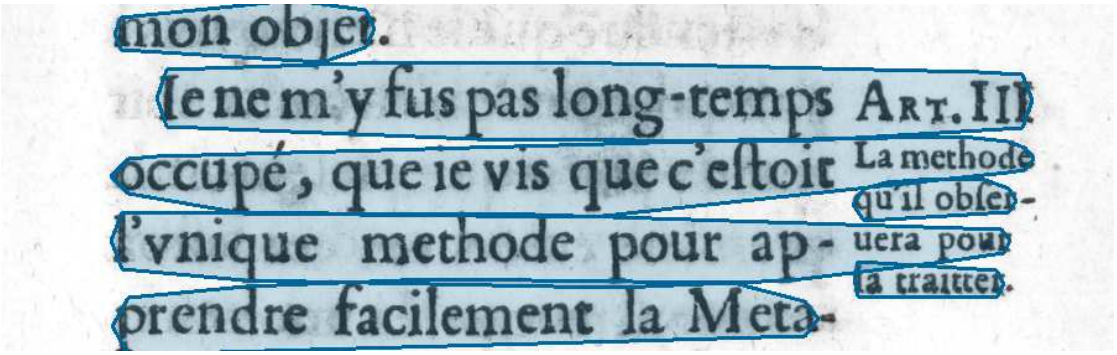


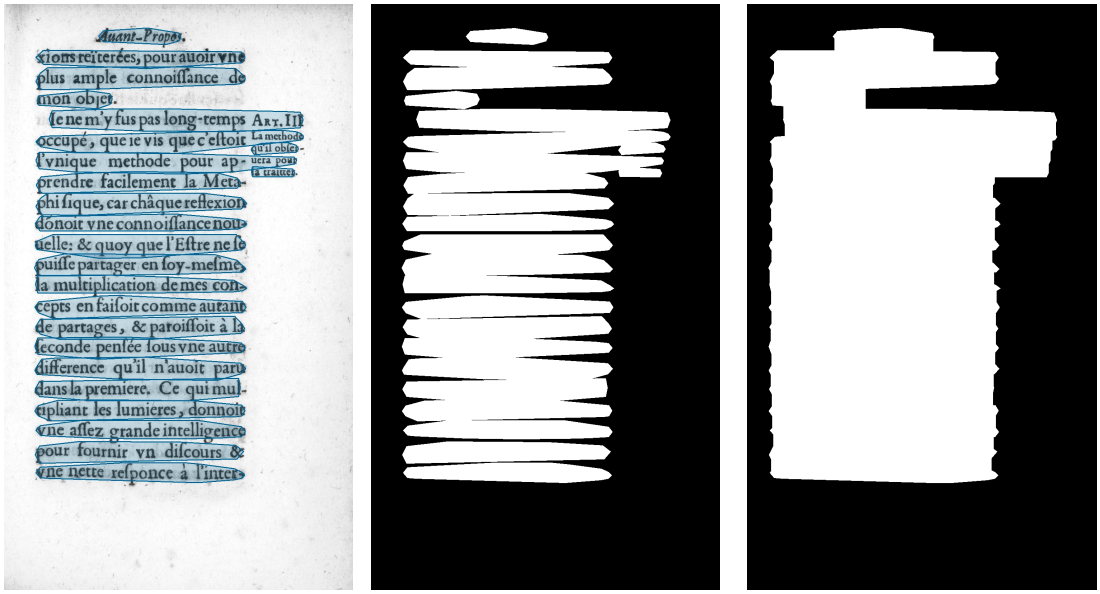
Figure 3.12: Document fragment showing text line regions merged with marginalia regions. Detected text lines are illustrated using blue polygonal regions.

Marginalia regions located close to text lines can lead to problems in the text line extraction. The marginal notes might be falsely grouped with nearby text lines causing subsequent errors in the text block formation. Figure 3.12 shows an example where text lines are merged with marginalia regions due to their proximity and similar font size. Due to these errors the segmentation of marginalia impacts the overall segmentation performance. In order to improve the ability to handle marginalia regions an additional processing step is proposed that aims to identify elongated narrow gaps between text lines.

The segmentation of marginal regions is based on local projection profiles combined with a text region mask derived from the previously extracted text lines. First the average skew angle of the extracted text lines is determined and all text regions are rotated by the negated average angle such that the text regions are aligned horizontally. Then a binary image is generated that contains the convex hulls of the text lines. This image is used to generate a text region map by applying a morphological closing followed by a

dilation. The closing merges neighbouring text lines and the dilation extends the regions in horizontal direction. The resulting binary mask is used to restrict the area where gaps should be detected.

The extraction process of the text region map is illustrated in Figure 3.13. The illustrations are based on the same document image that is also depicted in Figure 3.12. Figure 3.13a shows the whole document image and the initial text line regions coloured in blue, Figure 3.13b shows the corresponding binary text line regions and Figure 3.13c shows the final text region map created by applying morphological operations.

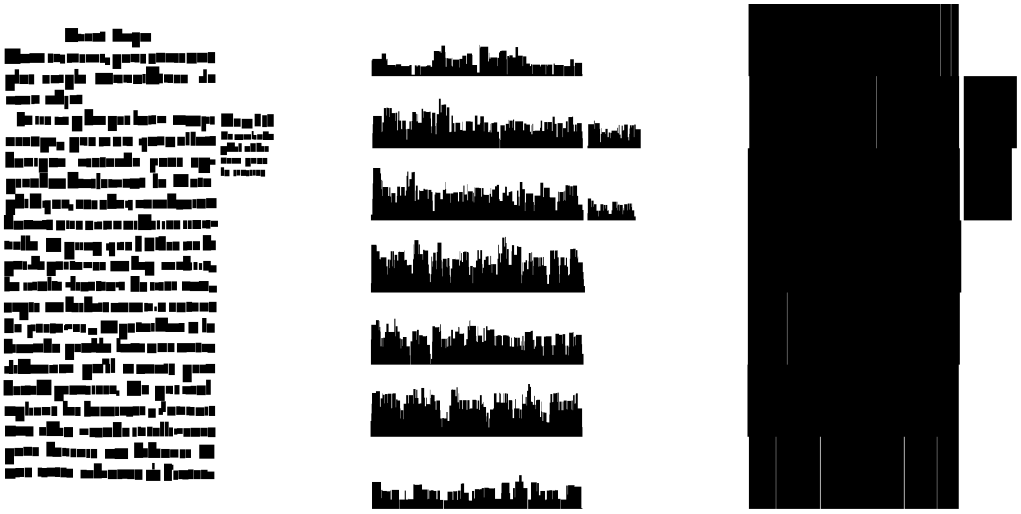


(a) (b) (c)

Figure 3.13: Illustrations showing the text region map extraction used for marginalia segmentation. Figure (a) shows the initial document image including detected text lines, (b) shows the binary image of the text line regions and (c) shows the extracted binary text region map.

In order to detect vertical gaps local projection profiles are used to generate a map of white spaces. For this purpose, another binary map is created that contains the BB areas of the text regions associated with the extracted text lines. An illustration of such a map is shown in Figure 3.14a. Then this map is split into horizontal strips and the vertical projection profile of each strip is computed. The resulting profiles are illustrated in Figure 3.14b.

A crucial parameter used in this step is the height of the strips. If the height of the strips is too small a lot of short gaps within the text blocks are extracted. On the other hand small marginals can be missed if the height is too big. Therefore, the height of the strips is set to approximately three times the text line height (see Section 3.1.1). This value represents a trade-off that reduces the amount of false detections but identifies elongated



(a) (b) (c)

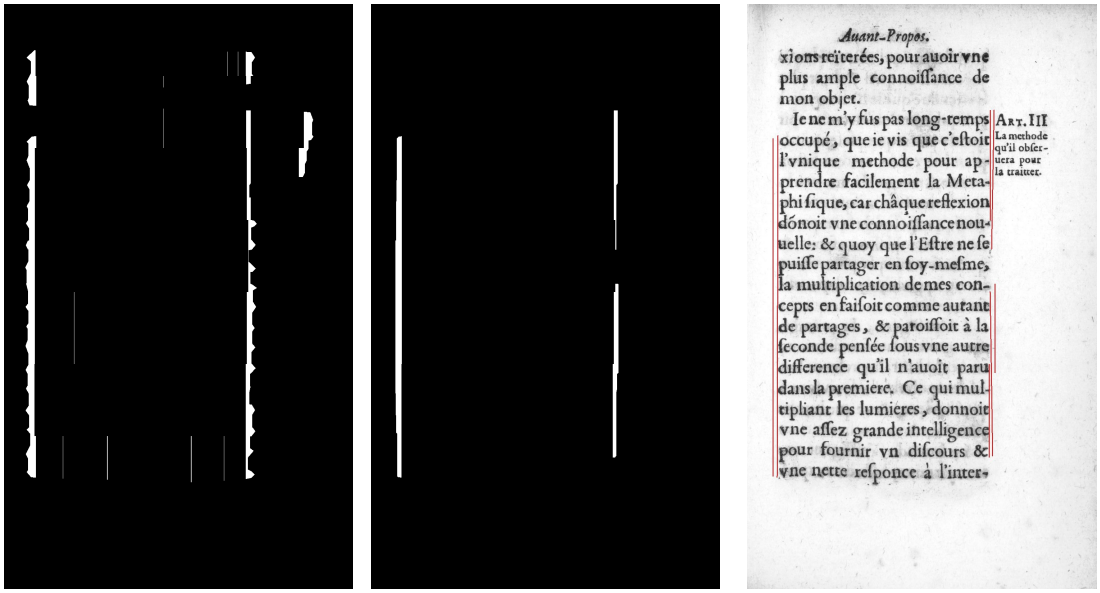
Figure 3.14: Illustrations showing the extraction of the map containing the white space gaps. Figure (a) shows the binary map containing the text region BBs, (b) shows the projection profiles of the strips of the binary map and (c) shows the resulting map containing potential white space gaps.

white separators. The map of white spaces gaps is then created by combining the zero points of all strips. An example of this map is shown in Figure 3.14c.

Due to the fact that only gaps near or between text line regions should be extracted the white space map is multiplied by the precomputed text region map. The final white spaces gaps are then found by applying a morphological closing that removes gaps that are narrower than five pixels or shorter than three times the text line height. Similar to the black separator detection the LSD algorithm is then used to extract separator lines from the white space map and split the text lines accordingly.

Figure 3.15 shows the final steps of the extraction process of the white space gaps. The combination of the white space map and the text region map is shown in Figure 3.15a. Based on this image the refined white space map is shown in Figure 3.15b and the final white separator lines are indicated with red lines in Figure 3.15c. The influence of the detected separators on the text lines is illustrated in Figure 3.16. Text line are depicted using blue polygonal regions and the separators are indicated with red lines. The image shows that the marginalia regions originally merged with the neighbouring text block in Figure 3.12 are correctly split off.

The impact of the marginalia segmentation on the segmentation performance is shown in Table 4.4 in Section 4.1. The additional identification of white space gaps results in a



(a) (b) (c)

Figure 3.15: Illustrations showing the extraction of the final separator lines used for marginalia segmentation. Figure (a) shows the combination of the text region and the white space map, (b) shows the refined white space map and (c) shows the final separator lines superimposed on the original document image.

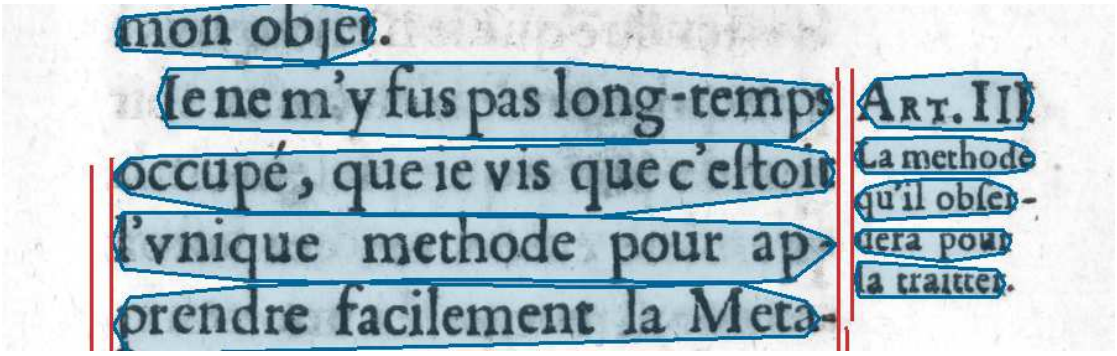


Figure 3.16: Results of the marginalia segmentation for a document fragment. Text lines illustrated using blue polygonal regions and white separators using red lines.

better ability to deal with the marginalia contained in the HBR2013 test data set. By applying the proposed marginalia segmentation the performance is increased by 3.33%.

As mentioned before marginalia regions occur in the documents contained in the HBR2013 data sets. For data sets that are less likely to contain marginalia or narrow white separators this processing step can be skipped. Applying the marginalia segmentation might produce erroneous splits, especially for documents with large character or word spacing. However, the results in Section 4.1 show the positive influence of applying the marginalia segmentation on the HBR2013 test data set.

3.1.3 Text Block Formation

Text blocks are formed by grouping text lines according to their adjacency relations and individual properties. The extraction of text block regions is crucial for the evaluation of the page segmentation methodology based on the HBR2013 data set and evaluation scheme. Expected results for the HBR2013 are bounding polygons for each homogeneous region of text. This means that paragraphs, headlines, annotations and other document regions are each represented by separate region polygons. Therefore, in this section the text block formation and the computation of the corresponding bounding polygons is discussed in detail.

Text Line Grouping

The text line grouping process is again inspired by the work from Chen et al. [CYL13]. The basic idea is to find the Below Nearest Neighbour (BNN)s for each text line and use this adjacency information to form chains of text lines, i.e. text blocks. BNNs are found by sorting the text lines in descending order and linking each line with other horizontally overlapping ones. For each line the area beneath it is investigated up to a distance of 1.5 times the text line height. Then the nearest linked line and further text lines that are vertically overlapping with the nearest line are selected to form the final list of BNNs. Figure 3.17 shows an example of text lines and their corresponding BNNs linked with blue lines.

A modification from the original method is an additional constraint that is imposed in the BNN linking process. In the proposed system text lines are linked with each other only if the following condition is fulfilled:

$$min(m_{h1}, m_{h2})/max(m_{h1}, m_{h2}) > 0.75 \tag{3.8}$$

where m_{h1} and m_{h2} are the median text region heights of the linked text lines. This condition ensure that the height of the text lines is similar. Applying this condition avoids false links between text lines from different document regions, e.g. links between headings and text body.

In the next step text blocks are found by linking text lines together based on the computed BNNs. The document image is processed in top-down order. Starting from the topmost

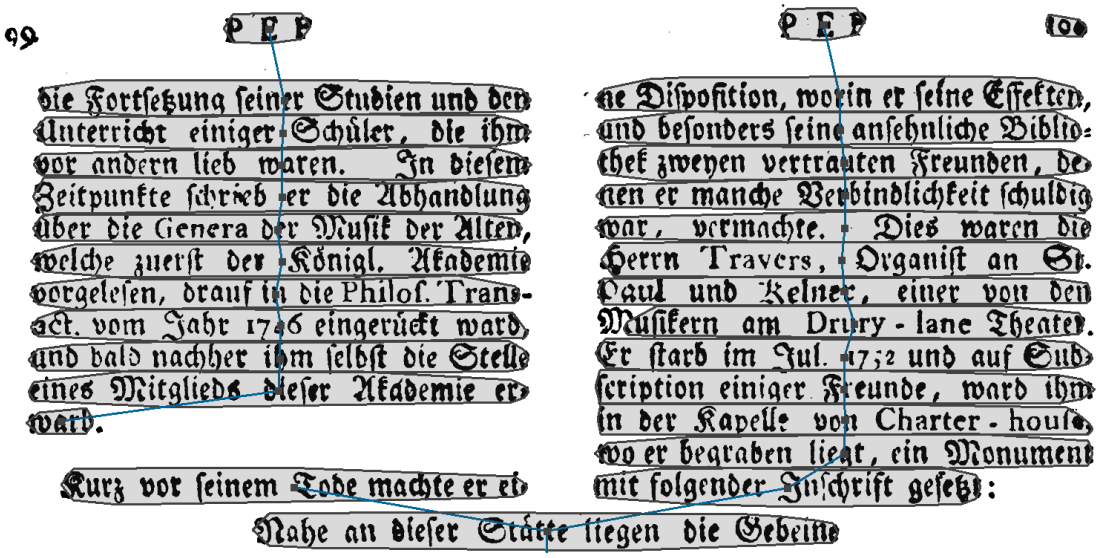
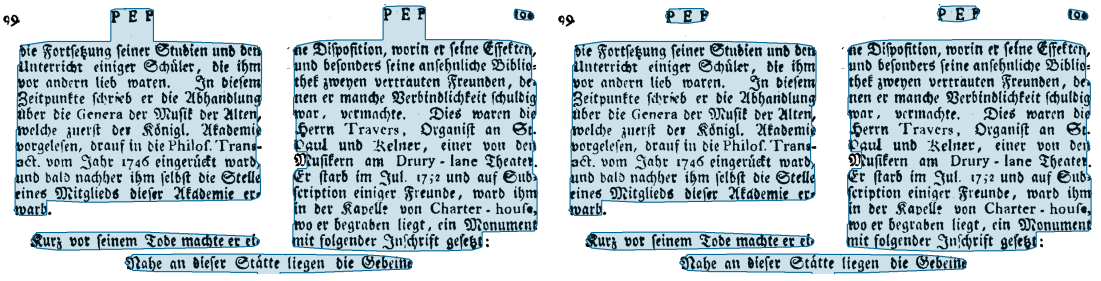


Figure 3.17: Text lines and their linked BNNs used for the text block formation.

text line a linked BNN is grouped if the number of BNNs is exactly one. This means that the grouping process is stopped if a text lines has no or more than one BNN. A text line with more than one BNN is assumed to be followed by multiple columns. Similarly, the grouping process is also stopped if multiple text lines link to the same BNN. The described grouping process is repeated until all text lines are processed and no further grouping is possible. In Figure 3.18a the results of this grouping process are illustrated for the same document fragment that is also depicted in Figure 3.17.



(a) (b)

Figure 3.18: Extracted text blocks of a document fragment before (a) and after (b) splitting them according to the median interline distance.

After the initial extraction of text blocks a further refinement is applied that splits the groups further. This refinement is designed to split text blocks at big gaps that occur at between paragraphs or near headings. In order to identify theses gaps the median interline distance is computed for each text block. Where the interline distance is computed as the distance between the base lines of two consecutive text lines. Then the text blocks

are split at gaps that are wider than the median interline distance increased by 20%.

An example of the final text blocks, after splitting according to the median interline distance is illustrated in Figure 3.18b. The figure shows that the refinement of the text blocks splits off the header regions from the paragraph blocks. However, the text block formation utilizes heuristic rules that are based on assumptions about the document layout. For example, the assumption that paragraphs are delimited by increased vertical gaps. Documents with more complex or irregular layouts violate these assumptions. Therefore, the performance of the text block formation is influenced by the document layout and the errors of the previously identified text lines.

Text Block Polygon Computation

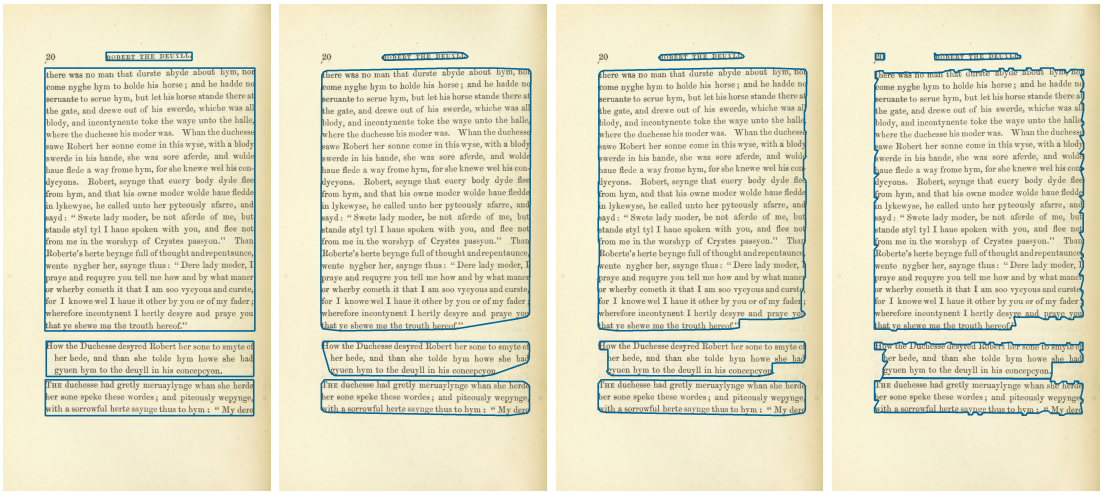
The segmentation performance of the text block formation stage is not only influence by the grouping process but also by the computation of the bounding polygon for the text blocks. According to the organizers of the HBR2013 [ACPP13a] each document region is described by a closely fitting polygon. Therefore, computing bounding rectangles for the found text blocks is an inaccurate approximation of the ground truth regions with respect to the evaluation scheme of the HBR2013.

Rather than using the BB of the text blocks the convex hull of the text lines is another alternative but it does not result in a closely fitting polygon. In the proposed system the bounding polygon is computed by connecting the convex hulls of consecutive text lines. The consecutive lines are connected with vertical lines starting from the inner bounds (i.e. maximum left and minimum right coordinate) of the lines. Then the bounding polygon is computed by finding the outer contour of the resulting region.

In Figure 3.19 the BB (3.19a), the convex hull (3.19b) and the proposed contour polygon (3.19c) for the text blocks of a document image are shown and compared with the ground truth (3.19d). The used document image is part of the HBR2013 training data set. As can be seen from the images the proposed contour polygon approximates the ground truth polygons best.

The evaluation results for the HBR2013 test data set shown in Table 4.5 in Section 4.1 show that the segmentation performance varies between 75.82% - 77.15%. The BB polygons with 77.15% result in the best performance followed by the convex hull polygons with 76.52% and the worst performance with 75.82% is achieved by using the tightly fitting contour polygons. The reason for this is that the coarse polygons are bigger and therefore merge regions that are erroneously split off or missed in the page segmentation process. As a result, the partial miss and split errors are reduced using the coarser polygons. On the other hand, the merge errors are clearly increased. Due to the evaluation profile the increased partial miss errors outweigh the merge errors. A more detailed analysis of the different polygon types is given in Section 4.1.

However, assuming a correct text line extraction the contour polygon, as shown in Figure 3.19, approximates the segmented text areas best. In addition, complex nonrectangular



(a) (b) (c) (d)

Figure 3.19: Images illustrating different kind of bounding polygons for text blocks in a document image. Figure (a) illustrates simple BBs, (b) shows the convex hull polygons, (c) shows the proposed contour polygon and in Figure (d) the corresponding ground truth polygons are shown.

layouts and cannot be accurately described using the rough polygon types. Therefore the coarse polygons types can result in a significant drop of segmentation performance for complex document layouts due to inevitable misclassification errors. Consequently the contour polygon is used as standard polygon type in the text block formation despite its inferior segmentation performance on the HBR2013 data set.

3.2 Font Style Classification

In this section the extraction of font style information from text regions and how it is integrated into the proposed recognition system is described. The font style classification is used to derive logical layout information based on the assumption that the logical role of the text regions can be determined based on their font style characteristics.

In order to extract font characteristics a texture analysis approach based on Gabor filters [Tan92] is proposed. A Gabor filter bank is used to extract feature vectors. These features are then used to train a classifier with labelled ground truth data that represents text regions with varying logical roles. Next input regions are processed and the extracted texture features are classified according to the trained font styles. The result of this process is a set of labelled text regions that can be used to determine the logical structure of the processed document image. The proposed font style classification method is designed to process text regions representing single words and classify them into a limited set of trained font styles.

In Section 3.2.1 a detailed description of the extraction of Gabor features is given. The classification and training process based on the extracted Gabor features is described in Section 3.2.2.

3.2.1 Gabor Feature Extraction

The use of Gabor filters for the extraction of texture features is already described by T. N. Tan [Tan92]. Tan states that it is not necessary to uniformly cover the entire frequency space for the task of texture discrimination. Hence, Tan proposes to use a set of Gabor filters where each filter is tuned a specific band of frequency and orientation. Based on the assumption that textual regions in a document image represent areas of distinct textures the approach by Tan can also be used for text region segmentation and classification. In addition, Mehri et al. [MGKH+13] show that Gabor features can be used to discriminate text regions of different fonts in historical document images and outperform GLCM as well as the autocorrelation function in terms of texture segmentation. As stated by Mehri et al. Gabor filters are an effective way to identify homogeneous regions in a document and do not rely on *a priori* knowledge such as typographical parameters. Therefore, in this work a Gabor filtering approach based on the principles proposed by T. N. Tan [Tan92] is used to classify word regions extracted from the historical catalogue images.

In Tan's work the feature extraction is done using a model that is designed to mimic the Human Visual System (HVS). The proposed model is based on the assumption that the HVS consists of parallel cortical channels that are tuned to signals with a specific band of spatial frequency and orientation. The fundamental elements of these channels, so called simple cortical cells, can be modelled by Gabor filters. In addition, it is assumed that two adjacent simple cells within a cortical channel have receptive field profiles of opposite symmetry. This means that two neighbouring simple cells can be modelled using a pair of isotropic Gabor filters with even and odd symmetry. According to these assumptions and some additional mathematical transformations Tan proposes the following computational model of the visual cortical channels:

$$\begin{aligned} q_e(x,y) &= h_e(x,y) * I(x,y) \\ q_o(x,y) &= h_o(x,y) * I(x,y) \\ q(x,y) &= \sqrt{q_e^2(x,y) + q_o^2(x,y)} \end{aligned} \tag{3.9}$$

where q_e and q_o denote the responses of the even and odd symmetric Gabor filters convolved with the input image I. These two-dimensional Gabor filters are composed of an isotropic Gaussian function and a sinusoidal plane wave that modulates it. The even and odd symmetric filters are defined as follows:

$$h_e(x,y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \cos(2\pi f(x\cos\theta + y\sin\theta)) \tag{3.10}$$

$$h_o(x,y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \sin(2\pi f(x\cos\theta + y\sin\theta)) \tag{3.11}$$

where the spatial frequency f, the orientation θ and the spatial constant σ are the parameters that define the response of the Gabor filters. By varying the parameters f and θ the receptive fields of different cortical channels are modelled. This equals to the extraction of texture features at varying scales and orientations.

Based on these definitions several methods [ZTW01, MD03, MGKH+13] have been proposed that use Gabor filtering to extract texture information at different levels of text. In the recognition system proposed in this work font style classification is computed at word level in order to segment input text lines according to their font style. Furthermore, the classification set of predefined font styles is done using manually annotated ground truth regions. Therefore, the font style classification component is inspired by the multi-class classifier proposed by Ma and Doermann [MD03].

The multi-class classifier proposed by Ma and Doermann [MD03] is used to classify scripts, font-faces and font styles at word level. In this work the set up by Ma and Doermann is adopted and used to distinguish word images written in Roman scripts based on their font style. In the context of this thesis font style classification refers to the identification of a predefined unique font style, where a font style is defined by its font face, style (regular, **bold**, *italic*, ***bold italic***) and size.

Gabor Filter Design

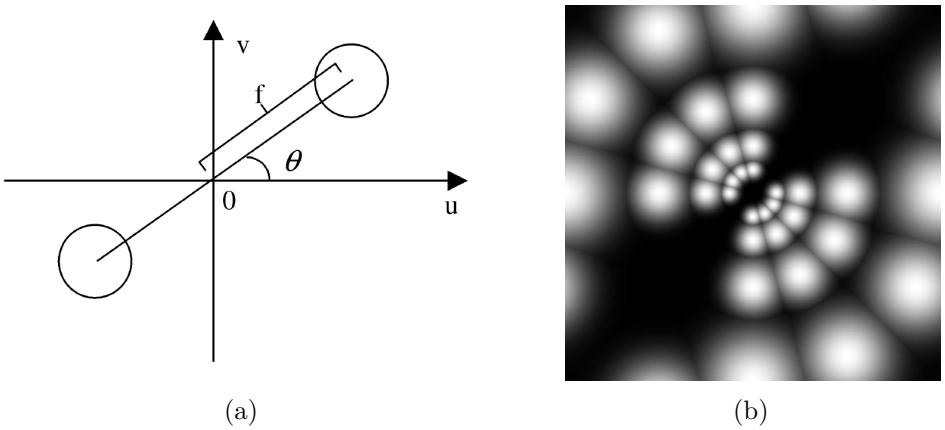
For the extraction of texture features it is crucial to design a set of Gabor filters that is able to discriminate different font styles but at the same time reduces the amount of redundant or ambiguous information that is extracted. Therefore, some guidelines are used to set the parameters f, θ and σ that define the set of Gabor filters. According to the experiments by T. N. Tan [Tan92] it is not necessary to uniformly cover the entire frequency plane. Furthermore, Tan concludes that the most relevant frequency components in an image of size $N \times N$ are limited by $f \leq \frac{N}{4}$ cycles per image. Based on these guidelines and the settings used in literature [Tan92, ZTW01, MD03] various sets are considered for the choice of the central frequencies. The final choice of the set of central frequencies is based on the evaluation results presented in Section 4.2.3. The values f_j determined for the spatial frequency parameter are: 2, 4, 8, 16.

The values considered for the orientation parameter θ are also based on the assumptions mentioned before. In addition, due to the central symmetry of the Gabor filters only half of the frequency plane needs to be covered. The final choice of the orientation values is again determined experimentally by using an iterative process as presented in Section 4.2.3. The selected orientation values are θ_i: 0°, 30°, 60°, 90°. Finally, similarly to the other methods presented in literature the spatial constant σ is chosen inversely proportional to the spatial frequency values f_j. To be more precise the value of the spatial constant is computed using the equation: $\sigma = \frac{1}{1.4 \cdot f_j}$. The value of the additional

multiplier (1.4) is once more determined experimentally based on the results presented in Section 4.2.3.

In Figure 3.20a the influence of the Gabor parameters in the frequency domain is illustrated and Figure 3.20b shows the response of the filter bank defined by the chosen parameter settings. The response image shows that the frequency plane is not covered equally. Nevertheless, the performance evaluation on the test data sets shows that the chosen parameter set provides better recognition rates compared to the basic parameter settings (θ_i: 0°, 45°, 90°, 135°; f_j: 2, 4, 8, 16; $\sigma = \frac{1}{f_j}$). By optimizing the parameters using specific training data the general parameter setup is adapted to the font style classification domain.

In addition, Figure 3.20b also shows that the inversely proportional values of the parameters f and σ complement each other. The inverse relation of the parameters accounts for the outward distance between the magnitude peaks. The spatial constant σ defines the standard deviation and therefore the size of the isotropic Gaussian envelope that constitutes the Gabor envelope in the spatial domain. In the frequency domain σ determines the size of the peaks representing the magnitude responses, whereas the frequency f corresponds to the distance from the centre.



(a) (b)

Figure 3.20: Illustration of the influence of the Gabor filter parameters (a), courtesy by [ZTW01] and the frequency response of the Gabor filter bank used for extracting texture features (b).

Feature Vector Computation

The chosen set of values for the parameters f and θ result in a filter bank of 16 (4 orientations × 4 frequencies) filter pairs (even and odd symmetrical). This filter bank is used to compute output images for each parameter combination based on the computational model presented in Equation 3.9. The actual feature vector that is used for font style classification is then constructed by computing the mean and standard

deviation of each output image. The mean and standard deviation are computed as follows:

$$\mu_{i,j} = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} q_{i,j}(x,y)}{M \cdot N} \tag{3.12}$$

$$\sigma_{i,j} = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} (q_{i,j}(x,y) - \mu_{i,j})^2}{M \cdot N} \tag{3.13}$$

where $q_{i,j}$ denotes the output image for the Gabor filter defined by the parameter values θ_i and f_j and the values M, N denote the size of the processed image patch (125×125 pixels in the proposed method). Based on these equations and the chosen parameter settings the feature vector is constructed as follows:

$$v = [\mu_{0,0}, \sigma_{0,0}, \mu_{0,1} ..., \mu_{i,j}, \sigma_{i,j}] \tag{3.14}$$

This means that for each image patch that is processed a 32 (4 orientations $\times$ 4 frequencies $\times$ 2 features) dimensional feature vector is computed. This feature vector is the basis of the texture analysis. It is used to train the font style classifier and compute respective font style results for input text regions.

3.2.2 Font Style Training and Classification

The classification process of predefined font styles is performed on word level. Therefore, the training and classification of font styles is based on the extraction of texture features from word images. In this section the general methodology for processing cropped word images is described. The individual steps of this process are developed based on experiments with synthetic images. This means that the word images are synthetically generated and are not affected by visual artefacts , incorrect segmentation or other problems.

Nevertheless, the font style classification process is designed for the purpose of processing text regions from historical catalogue images. The adaptations and additional steps needed to process the historical images are described in Section 3.3.1. In this section the focus is on processing image patches containing single words without additional preprocessing steps.

Texture Generation from Word Patches

The proposed font style classification method operates on word level and therefore the input regions have varying width and height. This means it is necessary to generate normalized texture patches with a fixed size from the input regions in order to make sure that texture features are extracted under consistent conditions. Ma and Doermann [MD03] state that they use replication and scaling of word images for the generation

of normalized texture patches. This technique is also used in this work and verified experimentally. The recognition performance on the historical catalogue images is improved by using texture patches rather than extracting the texture features directly from the input text regions.

The method proposed in this work utilizes the basic assumption that the processed document images are acquired under consistent conditions and share a common predominant text line height. The predominant text line height is determined based on the height of the (ground truth) word regions that are used for training the classifier. The estimate is computed as the 95% quantile of the heights of the training samples.

The predominant text line height is then chosen as the *patch height* parameter which is used to adapt the height of the word images. In order to ensure consistent conditions for the extraction of texture information the height of all processed word images is adapted to match the patch height. This means that word images that are larger than the patch height are cropped and smaller images are enlarged by padding them. The cropping is performed by removing either the top or bottom row of the image based on which row contains the least amount of foreground pixels. As a result, the input word images are converted into a set of word patches with uniform height. Using a fixed patch height ensures that height differences in the word images are also reflected in the textures generated from them.

In the next step squared texture patches are generated from the fixed height word patches. Based on the patch height it is determined how many rows of word patches can be fitted into the texture. Then rows are generated by replicating, concatenating and wrapping the word images. Finally, the rows are evenly spread across the height of the generated texture patch. This process generates texture patches of a predefined size independent of the length of the initial word images.

In the proposed method 128×128 pixels is used as the default size of the texture patches. This size is determined experimentally based on the evaluation results presented in Section 4.2.4. Furthermore, the results in Section 4.2.4 also show that the introduction of a minimum number of two rows of words patches within a texture patch improves the performance for the fixed texture size of 128×128 pixels. Therefore, the height of word patches (i.e. the *patch height*) is limited to 64 pixels (128/2) and word images exceeding this limit are down scaled before processing them further.

An overview of the texture generation process is shown in Figure 3.21. The figure demonstrates the extraction of a word image from a text lines and the following steps used to generate a texture patch from it. In addition, the figure also demonstrates cases of word image that are larger or smaller than the patch height.

Feature Classifiers

The training and classification process of word images is based on the extraction of feature vectors as described in Section 3.2.1. In the training stage a classifier is trained using

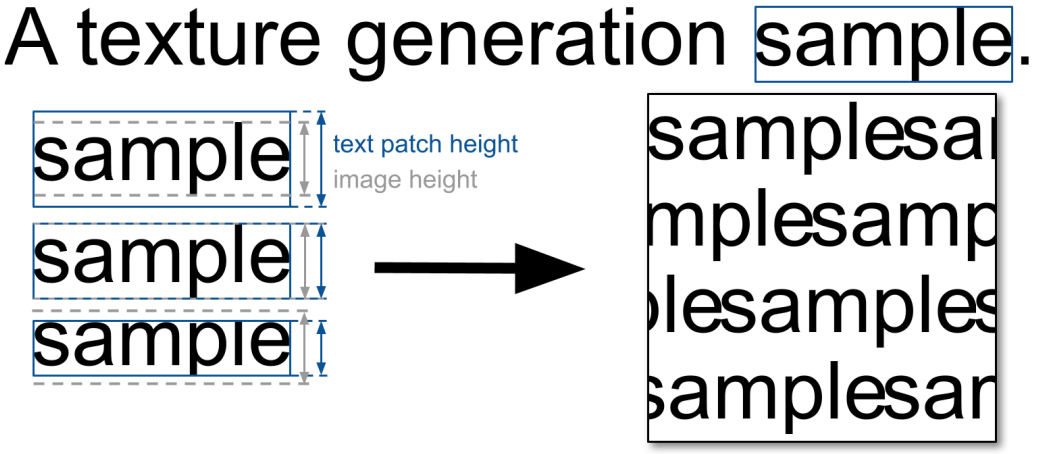


Figure 3.21: Illustration showing an overview of texture generation process from a word image extracted from a text line. On the left side the adaptation of the word image to different fixed patch heights is shown, and the resulting texture for the middle case on the right side.

word images representing a predefined set of font styles. After the initial training stage input word images are analysed and font style labels are assigned. The decision of the assignment is made according to which font style class results in the highest prediction probability.

The actual computation of the prediction probabilities depends on the type of classifier that is used. In the word level approach by Ma and Doermann [MD03], and the global texture analysis approach by Zhu et al [ZTW01]. a Weighted Euclidean Distance (WED) classifier is used. Zhu et al. state that any type of classifier can be used for classification but that the WED classifier is chosen for simplicity. In addition, the evaluation results by Ma and Doermann [MD03] show the effectiveness of combining the WED classifier in combination with Gabor features for a classification process on word level. They achieve a recognition accuracy above 73% for the tasks of script (two scripts at a time), font style (referring to bold, italic, normal) and font face (Arial, Times New Roman) classification. Therefore, the WED classifier is also used in the proposed font style classification method and compared with two other classifiers.

The classification results for the WED classifier are based on the following distance computation:

$$d_{WED}(v^k) = \sqrt{\sum_{i=1}^{N} \left(\frac{v_i - v_i^k}{\sigma_i} \right)^2} \tag{3.15}$$

where v is the input feature vector and v^k is the feature vector representing the centroid of the font style class k. The distance between those vectors is then computed as the sum

of the squared differences between each feature v_i, with $i = 1..32$, of the vectors which are weighted by the standard deviation σ_i of the corresponding feature over all training samples (from all font style classes). Based on this computation the font style of the input feature vector v is determined by finding the font style class k with the minimum distance.

The second type of classifier that is considered is a standard k-NN classifier [CH67]. The class of an input sample (vector) is determined based on the most common class among the set of the k nearest neighbours in the training set. Finally, the third classifier that is considered is a Bayesian classifier [Fuk90]. The used Bayesian classifier assumes that each class is normally distributed but does not necessarily expect that they are independent. Hence, the data distribution is described by a Gaussian mixture where each class is represented by one Gaussian.

Based on the results presented in Section 4.2.2 the Bayes classifier outperforms the two other types of classifiers. The k-NN classifier results in a performance of 75.1%. By varying the parameter k the performance of the k-NN classifier can be optimized but it is still inferior to the other two classifiers. The WED classifier provides recognition rates over 80% for synthetic test data sets containing four distinct font styles. The Bayes classifier is able to outperform the results of the WED classifier by approximately 10% and provides an average recognition rate of 89.7% over all evaluated data sets. Therefore, the Bayes classifier is chosen as default classifier in the proposed system and recommended over the other evaluated classifiers.

In addition, it should be mentioned that besides the choice of the classifier the size and content of the training data set can also influence the performance of the font style classification. As shown by the results in Section 4.2.4 words with a minimum length of five characters have higher recognition rates than short words (less than five characters). The short words regions sometimes contain an insufficient amount of texture features for a correct classification. This means that the classification performance of data sets or classes is lower the more short words are contained in them.

Furthermore, the size of the training data set is crucial for a stable classification performance. The evaluation results in Section 4.2.4 show that 100 samples per class suffice to achieve recognition rates above 94% on the evaluated synthetic data set. The use of more training samples improves the recognition rate further. However, in the case of training font styles contained in the historical catalogues the amount of training data that is used should be reduced to a minimum. The reason is that the generation of the training data needs to be done manually. Based on the evaluation results a minimum of 100 training samples per class is chosen as a trade-off between the number of samples and the recognition performance.

3.3 Historical Catalogue Image Processing

In this section the steps needed to compute the final results of the proposed recognition system are described. The system is based on combination of the page segmentation method presented in Section 3.1 with the font style classification method presented in Section 3.2. In order to apply these methods to the catalogues and combine them with each other they are adapted and the data flow between them is specified. Finally, for the computation of the actual recognition results Tesseract also needs to be integrated into system and combined with the other stages.

The process of adapting the layout analysis stages for the computation of the final results is described in Section 3.3.1. Subsequently, in Section 3.3.2 it is briefly explained how Tesseract is integrated into the system and used to create the final output of the recognition system.

3.3.1 Adaptation of Layout Analysis

The page segmentation methodology described in Section 3.1 is evaluated and adapted the requirements of the HBR2013. For processing the catalogue images the page segmentation process is simplified and the output is adapted the needs of the recognition system. This means that unlike the results used for the evaluation on the data set of the HBR2013 the output regions are not grouped into text blocks. Within the catalogue recognition system the results of the page segmentation stage are the text lines that are created in the initial text region grouping stage. For further processing in the other stages the text lines regions are simply described by their BB. The white space segmentation and the segmentation of marginalia are omitted when processing the catalogue images. The reason for this is that the layout of the catalogues is sparser compared to the book pages evaluated in the HBR2013 data set. Applying the additional segmentation steps would results in an oversegmentation of the catalogue pages due to the gaps and indents that are used to structure these documents. In Figure 3.22a an example of the page segmentation results for a catalogue image and the corresponding BB polygons 3.22b are shown. In addition, dependent on the actual catalogue minor changes to the parameters of the conditions used for linking the text regions might be required in order to correctly extract text lines. This is required for processing text lines that exhibit large variations of the font size (height ratio between consecutive letters > 3). However, using the default values is sufficient for correctly segmenting the majority of the analysed catalogues.

The extracted text lines are then passed on to the font style classification stage. For the training of this stage a minimum amount 100 manually segmented and labelled word regions for each font style is used. The font style classification is designed for processing word regions in order to be able to identify changes of the font style along a text line. This means that an additional segmentation of word regions is required for classification. In the proposed system the word segmentation is obtained by applying the Tesseract engine to the text lines and processing the resulting word regions. However, for the purpose of evaluation the ground truth regions of the words are projected to the text lines.

(a)

(b)

(c)

(d)

Figure 3.22: Images illustrating the combination of the page segmentation and font style classification stages. The polygonal results regions of the page segmentation (a) are converted to BB polygons (b). Then based on an additional word segmentation (c) the final font style classification results are computed and illustrated by coloured bounding boxes (d). Catalogue image used in the illustrations courtesy by [Arcc]

This represents a ceiling analysis for the performance of the font style classification on the text lines results. Illustrations of the projected word regions and their corresponding font style classification results are shown in Figure 3.22c and 3.22d. The evaluation of the combination procedure shows that the font style classification performance for all three tested catalogues data sets is above 85%. In fact, the performance for all evaluated font styles is above 90% except for the classes representing the identification numbers of the artworks.

3.3.2 Integration of Tesseract and Final Results

The integration of the Tesseract engine into the proposed system is straightforward. The same input image that is also presented to the layout analysis stage is used as input. Then the OCR results of the entire page are computed and mapped the text line regions computed by the page segmentation stage. Optionally regions that are missed or cropped by the page segmentation stage but are included in the Tesseract results can be added to the page segmentation results. The final results of the recognition system are then obtained by performing another mapping process. The results of the font style classification process are mapped to the text regions. This means that the text regions are labelled according to their font style, where a font style represents a specific category of information contained in the processed catalogue. Optionally regions that yield low prediction probabilities in the font style classification can be labelled as undefined and excluded from the final results. This avoids the inclusion of regions that are not formatted with the predefined set of font styles. The output of the system is a list of categorized text fragments.

In order to correct the errors of the individual stages an additional post processing stage can be used to refine the results. For this purpose, an additional classifier incorporating font style and geometrical features of the text regions can be applied. The combination of these properties can overcome issues caused by relying only on one type of feature. For example, geometrical properties can be used to correct erroneous font style labels of text regions based on their size or position within a line. However, in the proposed system the final results are verified manually.

CHAPTER 4

Results

4.1 Page Segmentation Results

In this section an evaluation of the proposed page segmentation method on the HBR2013 data set [ACPP13a] is presented. This data set contains historical document images from books that are similar to the application domain targeted by the proposed recognition system. Based on the results of the competition the proposed method is compared to commercial, open-source and state-of-the-art methods. Furthermore, the goal-oriented evaluation scheme used in the HBR2013 competition allows to evaluate the segmentation methods with respect to an OCR scenario.

In Section 4.1.1 the evaluation results on the HBR2013 data set are presented. In addition, a detailed analysis of the individual stages of the page segmentation method is presented in Section 4.1.2. Finally, segmentation results on the catalogue images are presented in Section 4.1.3.

4.1.1 HBR2013 Results

The proposed page segmentation method is evaluated on the HBR2013 test data using the publicly available Aletheia tool [ale]. The evaluation is based on combining weighted error rates as detailed in the competition summary [ACPP13a] and computing an overall success rate. The error types considered in the evaluation process are merge, split, miss (or partial miss), false detection and misclassification.

In the competition two different evaluation profiles are used to analyse the segmentation performance. The first profile is focused on pure segmentation and therefore disregards misclassification errors. The second profile is focused on text (OCR scenario) and therefore misclassification errors are included and misclassification of text is weighted highest. In Table 4.1 the results of the proposed page segmentation method (referred to as CatRec in the table) is compared with the results of the methods evaluated in the

HBR2013 competition. Details on the state-of-the-art methods (EPITA, JOUVE, PAL) listed in the Table can be found in the HBR2013 competition summary [ACPP13a].

	Tesseract 3	CatRec	FRE	EPITA	JOUVE	PAL
Segmentation Profile	67.2	75.83	80.6	80.6	86.9	91.4
OCR Scenario Profile	65.7	75.82	78.0	80.0	84.7	91.0

Table 4.1: Page segmentation results on the HBR2013 test data set. The proposed method (CatRec) is compared with other state-of-the-art methods evaluated in the HBR2013 competition.

The results show that the proposed page segmentation method outperforms Tesseract 3 but is inferior to the commerical ABBYY FineReader ®Engine 10 (FRE) and the state-of-the-art methods. However, it should be pointed out that the proposed text segmentation method is designed to identify text lines for further processing within the recognition system. This means that the focus of the segmentation is on the extraction of line regions rather than text blocks and no filtering of non-text regions is done. Therefore, the segmentation extracts only text regions and non-text regions such as separators, graphical regions and others are not included in the output.

The results of the page segmentation stage could be further improved by adding a non-text filtering stage and extracting the identified non-text regions. In the context of this thesis the main task of the page segmentation is to identify candidate text regions that are processed further by the subsequent components of the system. Thus, misclassified text regions are also analysed by the subsequent components and can be excluded later on in the system. In addition, the identification of the non-text regions is not relevant for the information extraction process. Therefore, unlike the performance on the HBR2013 test data set, the performance of the proposed recognition system only depends on the correct segmentation of text line regions.

4.1.2 Analysis of Page Segmentation Stages

In this section the individual stages of the page segmentation procedure, described in Section 3.1, are evaluated using the HBR2013 test data set and the corresponding evaluation scheme. Each stage of the page segmentation is used to address certain problems in the segmentation process. In order to demonstrate the usefulness of these stages varying setups are evaluated with respect to their segmentation performance.

Text Height Estimation

In Section 3.1.1 the process of text height estimation (THE) and the corresponding adaptive scaling is described in detail. The impact of applying text height estimation is shown in Table 4.2. In this table the results of the proposed segmentation method for applying two different text height estimation methods (Athena, rough) and applying no

text height estimation are shown. The results are computed using the HBR2013 test data set and the OCR scenario evaluation profile.

	no THE	Athena THE	rough THE
Overall success rate	74.2	75.3	75.8

Table 4.2: Page segmentation results using different Text Height Estimation (THE) methods and corresponding adaptive scaling.

Based on the results the rough text height estimation provides the best performance but the Athena method provides similar results. This means that by using the rough text height estimation and the corresponding adaptive scaling the segmentation performance of the proposed system is improved. Compared to the results of using no text height estimation the performance is 1.61% higher in terms of overall success rate using the text focused evaluation profile.

The advantage of applying text height estimation in combination with adaptive image scaling is that the candidate text regions are shrunk for images with higher resolutions. This reduces the size and amount of extracted candidate text regions. By reducing the image size the number of noise components or additional components created by oversegmentation in the MSER extraction process can be reduced. However, the maximum height parameter (50 pixels) is chosen such that only images with high-resolution text regions are down scaled. This means that only a subset of the images in the HBR2013 test data set is affected by the adaptive scaling. In addition, the MSER pruning stage, described in Section 3.1.1, makes implicit assumptions about the dominant text height and therefore reduced the impact of a missing or inaccurate text height estimation. The main purpose of the adaptive scaling is to avoid failures of the text region grouping process caused by an oversegmentation of text regions on high-resolution images. Therefore, the adaptive scaling stage is a useful addition to the recognition system despite its rather small influence on the overall segmentation performance on the HBR2013 test data set.

Text Region Grouping

The initial extraction of text lines using a grouping process of the extracted candidate text regions, as described in Section 3.1.2, results in a basic segmentation of text regions. This basic segmentation is then refined by applying further processing steps such as separator detection, white space segmentation and text block formation. In order to demonstrate the influence of theses stages a baseline performance is computed and used for comparison.

The baseline performance is represented by the evaluation results of the initial text region grouping stage with additional input image scaling (see Section 3.1.1). In Table 4.4 the results for the text region grouping (TRG) stage using the OCR evaluation profile and the HBR2013 test data set are shown.

	TRG	TRG+TBF	TRG+TBF+BSD
Overall success rate	56.4	69.6	70.2

Table 4.3: Page segmentation results of the initial text region grouping process (TRG) on the HBR2013 test data set. In addition, the results of combing the TRG with text block formation (TBF) and black separator detection (BSD) are shown.

The baseline performance without any additional improvements results in an oversegmentation with respect to the expected results for the HBR2013 evaluation scheme. The results regions need to be grouped into homogeneous blocks of text instead of single text lines to be in accordance with the ground truth regions. Therefore, the text block grouping stage is used as an additional step in order to provide comprehensible results. The baseline performance including additional Text Block Formation (TRG + TBF) is also shown in Table 4.4. Based on the results presented in Table 4.1 this refined baseline performance outperforms the Tesseract 3 segmentation. This demonstrates the efficiency of the basic stages of the proposed segmentation method but also points out the importance of the text block formation stage (+13.2%).

By using the Black Separator Detection (BSD) the baseline performance is further improved. The results (TRG+TBF+BSD) in Table 4.4 show that the black separator detection described in Section 3.1.2 improves the baseline performance by 0.58%. This improvement is small compared to other stages of the segmentation method. The reason for this is the small number of black separators in the HBR2013 test data set and the fact that some text line splits caused by black separators are detected by the text block formation without the additional information. The advantage of the black separator detection is that it produces hardly any false detections. Despite its rather small influence on the segmentation performance the separator detection increases the overall robustness and the gathered information can be easily incorporated in the other stages.

Next the refined basic setup (RBS = TRG+TBF+BSD) is used to investigate the use of white space information in order to improve the segmentation results. In Table 4.4 the results for applying additional white space segmentation and marginal segmentation are compared with the performance of the refined basic setup.

	RBS	RBS+WSS	RBS+WSS+MS
Overall success rate	70.2	72.5	75.8

Table 4.4: Page segmentation results of the refined basic setup (RBS) on the HBR2013 test data set. In addition, the results of combing the RBS with White Space Segmentation (WSS) and Marginalia Segmentation (MS) are shown.

Applying additional White Space Segmentation (RBS+WSS) increases the performance by 2.27%. The white space segmentation increases the ability to deal with more complex layouts by splitting text blocks into columns. However, as already mentioned in Section

3.1.2 the white space segmentation is not able to detect narrow white spaces gaps caused by marginalia or other regions. This is the reason why the additional marginalia segmentation stage is introduced. By applying both, white space segmentation and additional Marginalia Segmentation (RBS+WSS+MS) the segmentation performance is improved further by 3.33%. This means that overall the use of white space segmentation increases the performance by 5.6%.

The marginalia segmentation has larger influence on the performance than the white space segmentation. The reason for this is that marginalia are common in the HBR2013 data set. Furthermore, erroneous merges of marginalia regions cause subsequent errors in the text block formation. Due to these errors in the text block formation an increased number of erroneous text regions is created which negatively influence the performance.

Text Block Formation

Grouping the extracted text lines and computing a bounding polygon for the resulting text block regions has a strong influence on the evaluation results. As shown in Table 4.4 the text block formation accounts for the major part of the performance improvements based on the basic text line results.

An important aspect of the text block formation is the computation of the bounding polygons. In Section 3.1.3 three polygon types are presented that are considered for the computation of the bounding polygons. The detailed results for each of the three polygon types are shown in Table 4.5. The results are again computed using the HBR2013 test data set and OCR scenario evaluation profile.

	Contour Polygon	Convex Hull	BB
Overall success rate	75.8	76.5	77.2
Merge error rate	20.4	25.9	30.5
Split error rate	22.5	21.8	19.8
Miss error rate	6.7	4.8	5.7
Partial miss error rate	30.2	24.2	14.6
Misclassification error rate	1.4	1.9	2.5
False detection error rate	0.1	0.1	0.1

Table 4.5: Detailed error rates for the three types of bounding polygons considered in the text block formation stage.

The results show that coarser polygon types (convex hull and BB) are able to outperform the proposed standard polygon type (contour polygon) in terms of the overall success rate of the HBR2013 evaluation scheme. The reason for this is that the miss and partial miss errors are weighted highest in the used evaluation profile. By using BB polygons the partial miss error rate is halved compared to the contour polygon. In addition, the miss and split error rates are also slightly reduced. On the other hand, the biggest disadvantage of the BB polygons is the increased merge error rate (+10%).

Using BB or convex hull polygons increases the size of the extracted text block regions and therefore encloses regions originally not covered by the initial text line regions. In case of single characters or words that are erroneously split off from text lines this inclusion improves the segmentation results. The additional background pixels covered by the bigger polygons do not affect the performance as those pixels are excluded from weighting the segmentation errors. A drawback of the increased polygon area is that also other unintended region merges occur. This is also the reason why the misclassification error rate is increased.

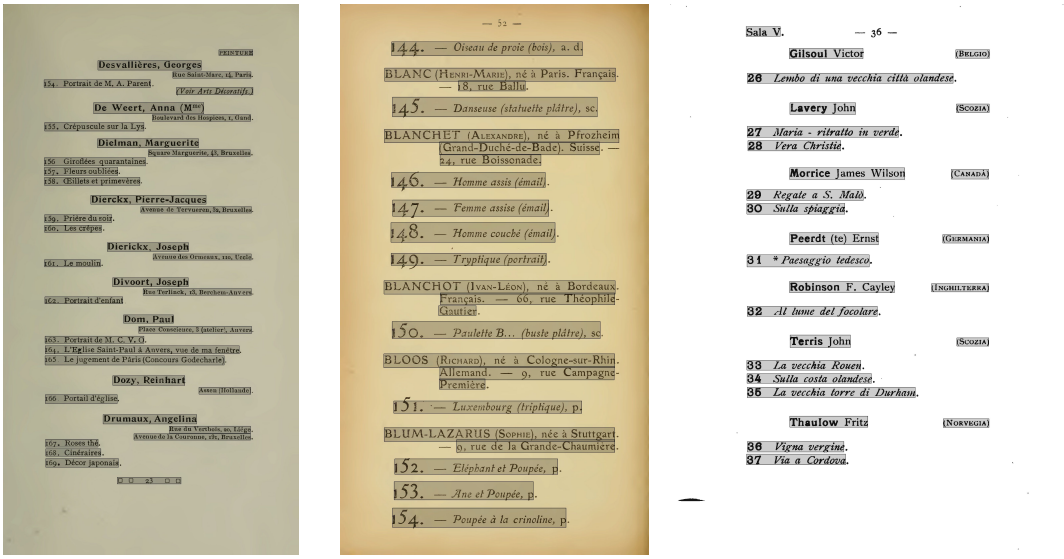
Due to the unintended merges and coarse representation of the text block regions the BB polygon is considered inappropriate for documents with complex layouts. Similarly, the convex hull polygon is also considered inaccurate. In other words, the contour polygon approximates the shape of the ground truth polygons best. Therefore, the contour polygon is used as standard polygon type in the text block formation. The lack of segmentation performance for this polygon type is caused by errors in the segmentation results. This, means that rather than choosing another polygon type the segmentation results should be refined using post-processing methods. For example, an additional region growing can be used to merge missed text regions near extracted text blocks, or an additional non-text filtering process can be used to reduce misclassifications.

4.1.3 Historical Catalogue Results

When processing the historical exhibition catalogues the page segmentation methodology is simplified compared to the full-scale approach used for the evaluation on the HBR2013. The adaptations of the page segmentation are described in Section 3.3.1. The foremost adaptation is that instead of blocks lines of text are extracted for processing the catalogue images. In addition, the area of the text lines is simply indicated by a BB.

Exemplary results of the page segmentation stage on three different catalogues are shown in Figure 4.1. The results show that there are only minor segmentation errors. The catalogue *Salon d'automne* illustrated in Figure 4.1b represents the biggest challenge with respect to the page segmentation. The regions representing identification numbers are sometimes oversegmented. Nevertheless, in the illustrated sample the text region grouping does not fail completely but on some regions the upper and lower parts of the IDs are split off. On other pages in this data set the IDs are not grouped correctly due to failed text region grouping caused by oversegmentation. The image in Figure 4.1c shows another example of a segmentation error. The page number located at the top of the page is discarded because of the threshold for the minimum number of components per text line.

In general, the proposed page segmentation approach is able to correctly segment the majority of the text regions contained in the analysed catalogues. However, the idiosyncrasies of the layout and formatting of the text in different catalogues poses challenges that might require an adaptation of the text region grouping parameters.



(a) (b) (c)

Figure 4.1: Images illustrating the page segmentation results on three different art exhibition catalogues. The catalogues shown are (a) Exposition générale des Beaux-Arts - Bruxelles 1907 [Arcc] (b) Salon d'automne - Exposition de 1907 - Paris [Arcb] (c) VI. Esposizione d'Arte della Città di Venezia [Arca].

4.2 Font Style Classification Results

In this section evaluation results for the proposed font style classification method are presented. In order to provide a detailed analysis of the characteristics of the method two types of data sets are used in the evaluation. The first type of data sets consists of word images extracted from different historical catalogues and the second type consists of synthetic word images. The synthetic data sets are used to test the abilities to deal with specific font style characteristics that are not contained in the catalogue images.

In order to optimize the font style classification method a mixture of six different data sets, three catalogue data sets and three synthetic data sets, is used. Based on these data sets the parameters of the Gabor filter bank are evaluated and different classifiers are tested. Additionally, further synthetic data sets are used to evaluate the texture generation and the training process in detail.

In Section 4.2.1 the data sets used in the evaluation process are described in detail and in Section 4.2.2 the evaluation results for different classifiers are presented. The choice of the parameters used to define the Gabor filter bank for feature extraction is discussed in Section 4.2.3. In addition, in Section 4.2.4 a detailed analysis of the texture generation and the following font style training process is presented. Finally, in Section 4.2.5 the evaluation results for cropped word images from the historical catalogues are discussed in detail.

4.2.1 Evaluation Data Sets

In order to provide comprehensive results a mixture of varying data sets is used in the evaluation process. Using this data set mixture in the evaluation process makes sure that the evaluated method is not fitted to a single data set but rather able to deal with a wide variety of font styles and documents.

The catalogue data sets used in the evaluation process consist of word images extracted from three different historical catalogues (Salon des Beaux-Arts - Bruxelles 1907 [Arcc], Salon d'Automne - Paris 1907 [Arcb], Esposizione Internationale d'Arte - Venezia 1905 [Arca]). From each catalogue at least 10 pages are used to generate training samples of different classes. Where a class represents a set of words that share a common font style. The font style information is determined based on manually generated ground truth annotations of the catalogues. For each class a minimum of 100 training samples is generated but in some cases additional (i.e. more than 10) pages or parts of pages are used to reach the minimum sample number for each class. For the test data set another 5 pages are used to generate test samples. The data sets corresponding to the three historical catalogues are referred to as *Cat1, Cat2, Cat3*. In Figure 4.2(a,b,c) samples from all three catalogue data sets are shown along with some basic information for each class.

For the synthetic data sets words are extracted from the book *The Innocents Abroad* by Mark Twain. Then for a limited set of words synthetic images are created that represent different fonts with varying sizes, styles and font faces. The three synthetic data sets (SynStyle, SynFace, SynSize) each comprise 100 training and 750 test samples for each class. In the data set SynStyle four classes are contained that represent the font styles regular, bold, italic and bold italic while the font size (30 pixels) and typeface (Arial) are fixed. In the second data set SynFace the font face (Arial, Times New Roman, Georgia, Franklin Gothic Medium) is varied while the style (regular) and font size (30 pixels) are fixed. The third data set SynSize contains 5 classes representing varying font sizes (10, 20, 30, 40, 50 pixels) while the style (regular) and typeface (Arial) are fixed. By combining these three data sets the ability to discriminate fonts based on their basic characteristics (size, style, font face) is tested. In Figure 4.2(d-f) samples from all three synthetic data sets are shown along with some basic information for each class.

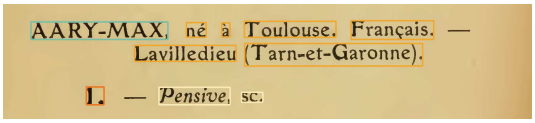
The mixture of the six previously described data sets is used in order to evaluate the classifier choice and the Gabor filter parameter settings in Sections 4.2.2 and 4.2.3. In addition, further synthetic data sets are used to evaluate further details of the classification process in Section 4.2.4.

4.2.2 Comparison of Font Style Classifiers

The choice of the classifier used to compute the prediction results has a major impact on the overall performance of the font style classification process. Therefore, three different classifiers are evaluated on the mixture of six data sets as described in Section 4.2.1. The performance of the classifiers is measured in terms of the recognition rate. The

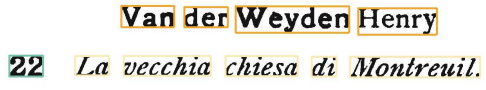
ID	description	samples train	samples test
1	artist	211	101
2	address	401	221
3	ID	158	84
4	title	521	260

(a) Cat1.



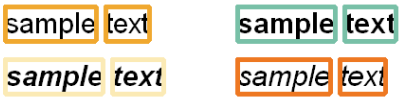
ID	description	samples train	samples test
1	artist	113	22
2	address	347	168
3	ID	114	50
4	title	423	219

(b) Cat2.



ID	description	samples train	samples test
1	artist	260	55
2	ID	97	90
3	title	288	246

(c) Cat3.



ID	description	samples train	samples test
1	normal	100	750
2	bold	100	750
3	bold + italic	100	750
4	italic	100	750

(d) SynStyle.

ID	description	samples train	samples test
1	Arial	100	750
2	Franklin Gothic M.	100	750
3	Times New Roman	100	750
4	Georgia	100	750

(e) SynFace.

ID	description	samples train	samples test
1	font size 10px	100	750
2	font size 20px	100	750
3	font size 30px	100	750
4	font size 40px	100	750
5	font size 50px	100	750

(f) SynSize.

Figure 4.2: Images of the catalogue (a-c) and the synthetic (d-f) data sets used for evaluation. The font style classes contained in each data set are illustrated using coloured boxes. Additionally, basic information about each class is provided in the table below the corresponding image.

recognition performance for a data set is computed as the average recognition rate RR over all classes of the data set:

$$RR = \frac{\sum_{i=1}^{N} RR_{c_i}}{N} \tag{4.1}$$

where RR_{c_i} is the recognition rate for class i and N is the number of classes. RR_{c_i} is defined as the number of correctly classified samples over the total number of samples of the class.

The simplest classifier type that is evaluated is the k-NN classifier. By varying the parameter k within the range 3-20 on the aforementioned data sets the optimal value is determined. The best results based on the average RR over all test data sets are achieved with the value $k = 5$. In addition to the k-NN classifier the performance of using just a single neighbour (i.e. a 1-NN classifier) is also computed and used as a baseline for the classifier evaluation.

The other two classifiers considered are a WED classifier and a Bayes classifier. The detailed evaluation results for the classifiers are presented in Table 4.6. In this table the RR for each of the six data sets and the average RR over all (avg), the synthetic (Syn) and the catalogue (Cat) data sets are listed.

	SynStyle	SynFace	SynSize	Cat1	Cat2	Cat3	Syn	Cat	avg
1-NN	63.6	55.5	79.0	68.3	77.4	79.7	66.0	75.1	70.6
5-NN	67.3	64.2	84.9	66.4	84.7	83.0	72.2	78.1	75.1
WED	74.5	67.5	88.9	75.3	79.3	86.6	76.9	80.4	78.7
Bayes	88.9	84.1	97.6	85.4	87.5	94.6	90.2	89.2	**89.7**

Table 4.6: Recognition rates for using different types of classifiers in percent. The columns 1-6 represent the results for single data sets. Additionally, the average recognition rates over the synthetic (Syn), the catalogue (Cat) and over all data sets (avg) are shown.

The results in Table 4.6 show that the Bayes classifier yields the best performance and outperforms the second best classifier, the WED classifier, by 11.0% in terms of the average RR over all data set. The third best classifier is the k-NN classifier and the 1-NN classifier achieves the lowest results. The baseline performance of the 1-NN classifier in terms of the average RR over all data sets is 70.6%.

With an average performance of 89.7% the Bayes classifier provides the best results on every single data set. Based on these results the Bayes classifier is chosen as the standard classifier in the proposed font style classification method. Therefore, the results presented in the following sections are all computed using the Bayes classifier.

4.2.3 Influence of Gabor Filter Parameters

The Gabor features used in the classification process change with respect to the parameters that define the Gabor filter bank. In order to optimize the performance of the proposed method the influence of three major parameters are evaluated. The three evaluated parameters are the spatial constant σ, the spatial frequency f of the sinusoidal component and the orientation θ.

Spatial Constant

First the parameter σ is evaluated. The authors of the multi-class Gabor filter approach [MD03] propose to set the parameter according to the following formula $\sigma = 1/(s \cdot f)$. Similar to other Gabor filter based methods [Tan92, MGKH+13, ZTW01] σ is computed as the inverse of the spatial frequency f but is additionally scaled by the multiplier $s = 0.6$. Based on this formula varying settings of s are considered and the value of σ is optimized. The other parameters θ ([0, 45, 90, 135]) and λ ([2, 4, 8, 16]) are set to standard values. In Figure 4.3 the evaluation results for varying the multiplier s are visualized in a plot.

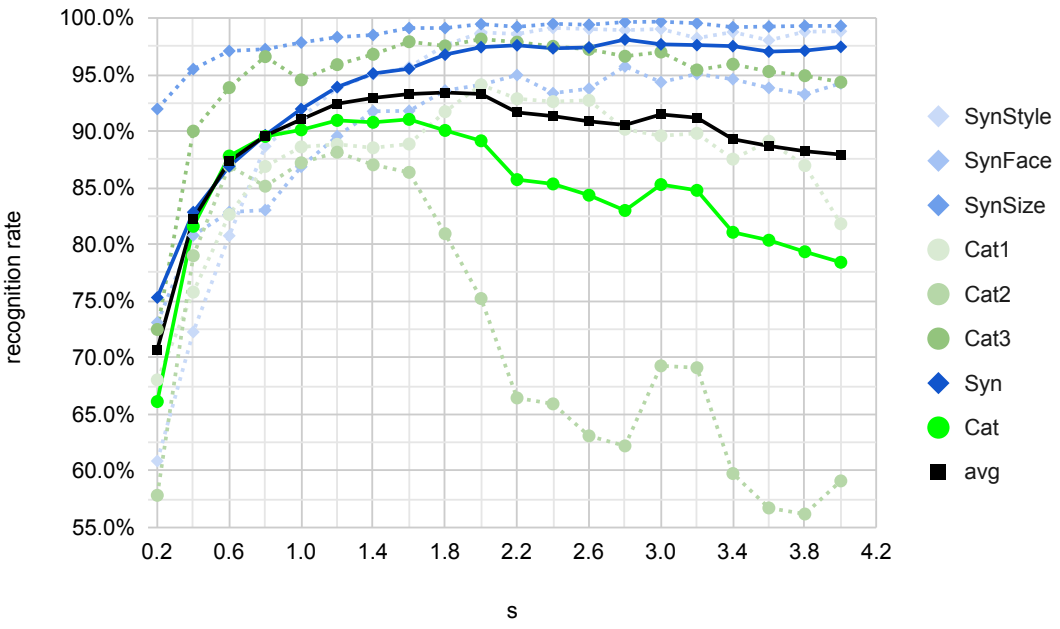


Figure 4.3: Chart visualizing the classification performance for varying the multiplier s of the parameter σ controlling the Gabor envelope size. Dashed lines indicate single data sets. The solid lines represent the average RR over the synthetic (Syn), the catalogue (Cat) and over all data sets (avg).

The plot shows the influence of the parameter σ on the synthetic and the catalogue data

sets. The average RR on the synthetic data set (Syn), indicated by the solid blue line in the plot, constantly increases for higher values of s in the range [0.2-2.0] attaining in a maximum performance of 97.4% for $s = 2.0$. For values in the range of s=2.0-4.0 the performance stagnates and varies between 97.0%-98.1%. On the catalogue data sets (Cat), indicated by the solid green line, the average performance also constantly increases for higher values of s in the range of [0.2-1.2] with a maximum performance of 91.0%. In the range [1.0-1.8] the performance stagnates and varies between 90.1%-91.1%. Unlike the synthetic data sets the performance drops for values of $s > 1.8$. The main reason for this drop is the performance on data set Cat2. The performance on Cat2 exhibits a large drop for values of s>1.6 and remains below 70% in the range [2.2-4.0].

The reason for the difference in the in the performance between synthetic and catalogue data sets is that the synthetic data sets do not contain noise, background variation and other artefacts. Based on the inverse relation in the formula $\sigma = 1/(s \cdot f)$ increasing the value of s reduces the value of σ. The value of σ controls the size of the magnitude peaks of the Gabor filters in the frequency domain. The higher the value of s the larger is the size of the peaks and therefore the sensitivity to image artefacts. In the data sets Cat1 and Cat2 the variation of the background is higher and more artefacts are contained compared to Cat3. Hence, the performance on Cat3 is more stable for higher value of s.

Based on the average RR over all data sets, indicated by the solid black line in Figure 4.3, the best performance with 93.4% is achieved for $s = 1.8$. However, in order to increase the robustness to image artefacts the default value of s is chosen based on the RR over the catalogue data sets. Therefore, the value of the parameter s should be chosen within the range [1.0-1.8] which provides a stable performance. By default, the value is set to $s = 1.4$ representing the middle of the range. This means that the value of σ is determined using the formula $\sigma = 1/(1.4 \cdot f)$.

Spatial Frequency

The next step is the optimization of the spatial frequency parameter f. This parameter does not represent a single value but a set of values that defines the Gabor filter bank which is used for feature extraction. Based on the upper limit $f \leq \frac{N}{4}$ (see Section 3.2.1) and the texture size of 128x128 pixels the following spatial frequency values are selected and evaluated: [2, 4, 8, 16, 32]. In addition, multiplying the values by $\sqrt{2}$, as proposed by Mehri et al. [MGKH+13], is also considered.

For a detailed analysis each of the individual frequency parameters is tested separately. The performance on the synthetic and the catalogue data sets for using each single frequency parameter in combination with $\sigma = 1/(1.4 \cdot f)$ and $\theta = [0, 45, 90, 135]$ is listed in Table 4.7.

The results in Table 4.7 show that the value 32 for the spatial frequency parameter f yields the worst performance. Increasing the spatial frequency means that the width of the strips in the Gabor kernels is enlarged. Therefore, higher values of f are better suited for large fonts, whereas smaller fonts require smaller central frequencies. Due to

f	SynStyle	SynFace	SynSize	Cat1	Cat2	Cat3	Syn	Cat	avg
2	82.2	72.8	83.5	59.0	71.3	85.6	79.5	72.0	75.7
4	78.8	69.0	92.1	73.1	78.0	83.2	80.0	78.1	79.0
8	59.0	63.6	91.8	78.4	82.8	81.8	71.5	81.0	76.2
16	43.9	45.0	82.0	75.0	73.2	68.5	57.0	72.2	64.6
32	43.3	43.5	74.8	49.7	50.8	56.9	53.8	52.5	53.2

Table 4.7: Recognition rates (in percent) for varying the parameter f controlling the width of the Gabor filter kernels. Columns 1-6 represent single data sets. The columns *Syn*, *Cat* and *avg* represent the average *RR* over the synthetic, the catalogue and over all data sets.

the fact that a limited texture size (128x128 pixels) containing at least two lines of text is used the maximum size of fonts contained in the processed textures is limited. The data set SynSize contains fonts of varying size and the largest font (50 pixels) almost reaches the maximum text patch height that can be fit into a 128x128 texture without down scaling. For $f = 32$ a *RR* of 74.8% is achieved on the data set SynSize. This is the best performance for $f = 32$ among all evaluate data sets but even on this data set all other frequency values outperform $f = 32$ by at least 7.2%. On the other data sets that contain smaller fonts the performance for $f = 32$ is only between 43.3%-56.9%. Therefore, it is concluded that $f = 32$ contributes most to data sets that exhaust the maximum font size but might reduce the performance on data sets that contain only smaller fonts.

In addition, the results in Table 4.7 also show that the lowest frequency values 2, 4 provide the highest *RRs* over the synthetic data sets (Syn). Whereas the highest performance on the catalogue data sets (Cat) is achieved for $f = 8$. The reason for this not only the font size of the text regions contained in the data sets but also the fact that smaller frequencies are more vulnerable to noise and image artefacts. Therefore, the performance of the lower frequency values is higher on data sets that contain less noise. This also explains why the catalogue data set Cat3, containing the least amount of noise, yields better results for $f = 2$ than the other catalogue data sets.

Based on the single value analysis the selected values of f are used and varying combinations of f are analysed in order to identify the optimal set of values for the parameter f. In Table 4.8 the results for various sets of values are presented.

The set [2; 3; 4; 6; 8] achieves the best results with an average *RR* of 93.9% over all data sets. The second best performance with 1.0% less is achieved by using the sets [2; 4; 8; 16] and [2; 3; 4; 6; 8]$\sqrt{2}$. Among these three sets [2; 4; 8; 16] provides the best performance on the catalogue data sets (90.8%) and covers a wider range of frequencies (i.e. a wider range of font sizes). In addition, the closer sampling, especially in the lower frequency range, for the set [2; 3; 4; 6; 8] is considered to be more sensitive to noise. Therefore, the set [2; 4; 8; 16] is chosen as the standard setting for the spatial frequency parameter f.

f	SynStyle	SynFace	SynSize	Cat1	Cat2	Cat3	Syn	Cat	avg
[2; 4; 8]	95.5	93.3	98.7	86.9	81.4	95.5	95.9	87.9	91.9
[2; 4; 8; 16]	95.0	91.8	98.5	88.5	87.0	96.8	95.1	**90.8**	92.9
[2; 3; 4; 6; 8]	98.2	95.7	99.6	89.3	85.2	95.7	**97.8**	90.1	**93.9**
[2; 4; 8; 16; 32]	93.3	90.0	98.1	88.7	84.8	98.8	93.8	**90.8**	92.3
[2; 3; 4; 6; 8; 12; 16]	96.5	92.4	99.2	92.4	73.8	90.1	96.0	85.4	90.7
$[2; 4; 8]\sqrt{2}$	89.7	90.9	97.4	88.7	87.8	93.8	92.6	90.1	91.4
$[2; 4; 8; 16]\sqrt{2}$	86.7	89.0	97.3	89.7	88.3	94.5	91.0	**90.8**	90.9
$[2; 3; 4; 6; 8]\sqrt{2}$	93.5	93.2	98.9	92.3	84.9	94.7	95.2	90.6	92.9
$[2; 4; 8; 16; 32]\sqrt{2}$	83.7	86.2	96.5	88.4	79.6	94.8	88.8	87.6	88.2
$[2; 3; 4; 6; 8; 12; 16]\sqrt{2}$	87.3	90.0	98.6	93.8	77.7	93.6	92.0	88.3	90.2

Table 4.8: Font style classification results for varying sets of the parameter f.

In addition, the results in Table 4.8 also show that applying the additional multiplier $\sqrt{2}$ does not have a significant positive influence on the average RR over all data sets. In fact the results show that the average RR for each parameter set is inferior compared to the sets without a multiplier.

Orientation

Finally, the orientation parameter θ is optimized. Based on an iterative analysis process a set of values is identified that optimizes the performance on the mixture of data sets described in Section 4.2.1. The first step of the process is the analysis of single values of the orientation parameter. Due to the central symmetry of the used Gabor filters only half the frequency plane needs to be covered. In the spatial domain the orientation value $\theta = 0°$ corresponds to a vertical alignment of the Gabor function. In Figure 4.4 the performance results for single θ values within the range [0°-165°] with a step size of 15° are plotted and additionally mirrored to visualize the entire range of [0°-360°]. These results are computed in combination with the previously found parameter values $\sigma = 1/(1.4 \cdot f)$ and $f = [2; 4; 8; 16]$.

The single value results shown in Figure 4.4 show that the value $\theta = 90°$, corresponding to a horizontally oriented Gabor filter, provides the best results in terms of average RR over all data sets. This means that despite the fact that most text strokes are vertical the extraction of horizontal strokes provides better results in terms of font style classification. One reason for the low performance at $\theta = 0°$ on the catalogue data sets is that this value provides well below average recognition rates for the classes representing the identification numbers. Therefore, the value $\theta = 90°$ is chosen as the basis for the following analysis steps. The value is fixed and combined with the remaining θ values within the range [0°-165°] in order to find the best set of two θ values in terms of average recognition performance. Then this procedure is repeated three times in order to find optimized sets of three, four and five values respectively.

In Table 4.9 the results for a selection of sets of the parameter θ are listed along with

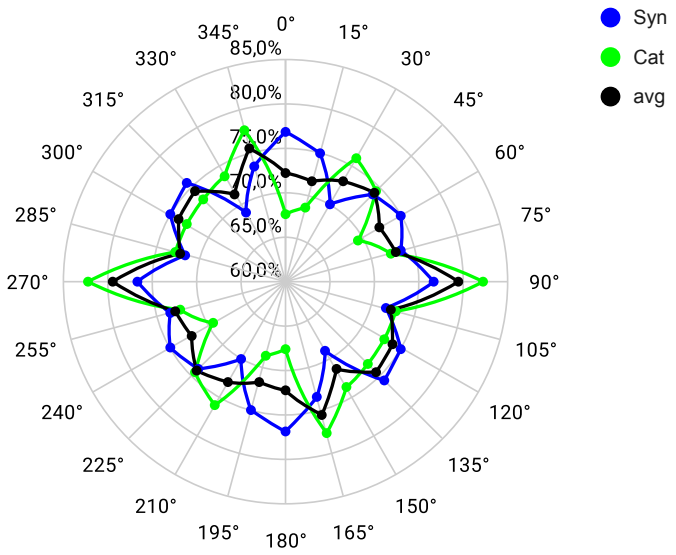


Figure 4.4: Plot illustrating the classification performance for individual values of the orientation parameter θ. The lines indicate the average RR in percent over the synthetic (Syn), the catalogue (Cat) and over all data sets (avg). The values $[180° - 345°]$ are mirrored for visualization.

the optimized sets of length three, four and five. The sets providing the best overall recognition performance are $\theta = [0°; 30°; 60°; 90°]$ and another set comprised of five values $\theta = [0°; 30°; 60°; 90°; 135°]$. Due to the fact that the former set consists only of four values and provides an equal RR over all data sets it is chosen as the standard set for the parameter θ.

Based on the results shown in Table 4.9 and the evaluation of further sets not listed in the table the following conclusion for the choice of θ are drawn. Sets of four and five values provide the best results. Sets of more than five values can result in performance drops presumably caused by overfitting. Furthermore, a uniform distribution of the values (e.g. $[0°; 45°; 90°; 135°]$) does not provide the best results with respect to the chosen parameter choices. The optimized set ($\theta = [0°; 30°; 60°; 90°]$) provides an improvement of 6.0% in terms of RR over all data set and outperforms the standard set on all six evaluated data sets.

One reason for this performance difference is the choice of the spatial constant σ. This parameter controls the bandwidth of the Gabor filters and therefore also the size of the magnitude peaks in the frequency plane. For the chosen setting $\sigma = 1/(1.4 \cdot f)$ the magnitude peaks of the set $[0°; 45°; 90°; 135°]$ do not touch each which means that the frequency information between the peaks is lost. The optimized set $\theta = [0°; 30°; 60°; 90°]$ represents a closer sampling with respect to the orientation parameter. The magnitude

θ	SynStyle	SynFace	SynSize	Cat1	Cat2	Cat3	Syn	Cat	avg
[0; 90; 60]*	93.1	90.2	98.1	89.7	88.0	95.2	93.8	91.0	92.4
[0; 30; 60; 90]*	94.4	90.8	98.5	90.1	91.9	97.2	94.6	93.0	**93.8**
[0; 15; 60; 90]	93.8	88.4	98.6	89.6	94.9	96.6	93.6	93.7	93.6
[0; 60; 90; 135]	95.4	92.5	98.4	89.9	85.9	98.6	95.4	91.5	93.5
[0; 45; 90; 135]	88.9	84.1	96.2	88.5	82.0	87.0	89.7	85.8	87.8
[0; 30; 45; 90]	87.1	83.3	94.1	86.7	82.8	84.1	88.1	84.5	86.3
[0; 30; 60; 90; 135]*	96.1	92.9	98.5	90.1	87.4	98.0	95.8	91.8	**93.8**
[0; 30; 60; 90; 120]	95.6	92.8	98.9	88.0	87.3	97.4	95.8	90.9	93.3
[0; 30; 60; 90; 120; 150]	88.7	81.1	91.5	89.8	75.4	65.4	87.1	76.9	82.0
[0; 22; 45; 67; 90; 112; 135; 157]	69.7	64.8	76.9	87.3	65.3	51.1	70.4	67.9	69.2

* optimized set

Table 4.9: Recognition performance for varying sets of the parameter θ. The values are the RR in percent and the marked sets represent optimized sets found by applying an iterative evaluation process.

peaks of the Gabor filter bank corresponding to this set slightly touch each other and cover one quadrant of the frequency plane (see Figure 3.20). Based the presented results it is concluded that the setting of the spatial constant σ and the orientation θ should be adapted to each other in order to optimize the performance. In addition, the results show that covering just one quadrant of the frequency plane provides sufficient texture information to achieve recognition results above 90% for the task of font style classification on synthetic and catalogue word images.

However, it should be pointed out that there are also other sets for the parameter θ that achieve a similar performance in terms of the RR over all data sets. The best choice for θ also depends on the typographical features of the actual font styles contained in the processed data sets. For example, italics are typically within the range $[0° - 15°]$. Therefore, adapting the orientation parameter set according to a specific data set can improve the performance even further.

Summary

Based on the evaluation results presents in this section the final parameter settings used for the computation of the font style classification results are:

- $f = [2; 4; 8; 16]$.
- $\theta = [0°; 30°; 60°; 90°]$
- $\sigma = 1/(1.4 \cdot f)$

In addition, the evaluation results show that the choice of the parameters σ and θ influence each other. Furthermore, it is also shown that the choice of the optimal parameters is

dependent on the presence of image artefacts and specific typographical features in the processed data sets.

4.2.4 Influence of Texture Generation and Training Data

In this section the influence of the texture generation and training data composition is analysed. First some details of the texture generation process are analysed and their impact on the classification performance is shown. Then the influence of the training data composition is investigated.

Texture Generation

The generation of textures from input word patches is a crucial step in the font style classification process. A basic parameter is the size of the textures that are generated. Based on the size of the input patches from the catalogue data sets and the values used in other state-of-the-art methods [MD03], [ZTW01], [MGKH+13], three different options are considered for the choice of the texture size: 64x64, 128x128, 256x256.

In Table 4.10 the results for different texture sizes on the synthetic and catalogue data sets are listed. The best performance with $RR = 93.8\%$ over all data sets is achieved with a texture size of 128x128 pixels. 256x256 provides the second best RR with 93.2% and 64x64 achieves the worst performance with $RR = 89.8\%$. On the synthetic data sets the texture size 256x256 achieves the best results (94.9%) outperforming 128x128 by just 0.3%. On the catalogue data sets the texture size 128x128 provides the best results (93.1%) outperforming 256x256 by 1.7%.

texture size	SynStyle	SynFace	SynSize	Cat1	Cat2	Cat3	Syn	Cat	avg
64x64	85.7	86.1	97.8	83.1	92.4	93.7	89.9	89.7	89.8
128x128	94.4	90.8	98.5	90.1	91.9	97.2	94.6	**93.1**	**93.8**
256x256	95.0	91.2	98.6	92.5	85.8	95.9	**94.9**	91.4	93.2

Table 4.10: Recognition rates in percent for using varying texture sizes on the synthetic and the catalogue data sets.

In order to provide a more comprehensive analysis of the texture size parameter further synthetic data sets are used. The additional synthetic data sets each contain four classes that represent the same four fonts (see SynStyle: Arial + 4 styles) but with a varying size (i. e. font size 10-100 pixels). These data sets are referred to as *FS10-FS100*, where the number refers to the font size. The recognition performance on the additional data sets is listed in Table 4.11. The results show that for the synthetic data sets the biggest texture size (256x256) provides the best results on each data set with an average RR of 95.1% over all data sets.

Based on the results it is concluded that bigger textures provide better or equal results for synthetic data sets. In addition, the results also show that the performance stagnates

data set	64x64	128x128	256x256
FS10	74.8	73.4	78.3
FS20	86.3	87.2	89.8
FS30	85.7	94.4	95.0
FS40	92.4	97.3	97.5
FS50	90.6	97.4	97.6
FS60	91.0	97.4	98.6
FS70	90.6	97.2	98.4
FS80	88.9	97.2	98.2
FS90	91.7	97.7	98.8
FS100	91.2	97.7	99.2
avg	88.3	93.7	**95.1**

Table 4.11: Font style classification results for using varying texture sizes on synthetic data sets containing samples with varying font sizes. The values represent the *RR* in percent.

and varies within a range of 2% for the data sets where the font size is bigger than half the texture size. The reason for this is that for these data sets the text patches are scaled down. Nevertheless, using a texture size of 128x128 provides a recognition rate over 90% for the data sets FS30-FS100. Based on the results presented for the synthetic and the catalogue data sets a fixed texture size of 128x128 pixels is used in the proposed font style classification approach.

As already mentioned, text patches are sometimes down scaled in the texture generation process. This is done in order to fit the text patches into the textures of 128x128 pixels. The scaling is performed with respect to the height of the text patches and a restriction of the minimum number of rows of patches per texture. By default, a minimum of two rows are fitted into a texture. The influence of changing the minimum number of rows is evaluated by computing the classification results for varying the parameter within the range 1-3.

In Table 4.12 the results for varying the minimum rows parameter are shown for the data sets Cat1-3 and FS40-100. Where FS40-100 are the same data sets as explained earlier in this section. The results for the data sets SynStyle, SynFace, SynSize and FS10-30 are omitted because they are equal for all three parameter settings due to their small text patch height. The results show that for the synthetic as well as the catalogue data sets the best performance is achieved by using a minimum row number of two. The average performance for two rows on the synthetic data sets is 97.4%, on the catalogue data sets the average performance is 93.0% and the average *RR* over all data sets is 95.2% Therefore the minimum number of rows per texture is set to two for the texture size of 128x128 pixels.

data set	1 row	2 rows	3 rows
FS40	97.3	97.3	95.2
FS50	97.4	97.4	95.9
FS60	97.9	97.4	94.0
FS70	97.5	97.2	94.7
FS80	96.0	97.2	96.7
FS90	97.3	97.7	93.9
FS100	97.2	97.7	94.0
avg FS	97.2	**97.4**	94.9
Cat1	90.1	90.1	89.6
Cat2	80.0	91.9	90.5
Cat3	95.1	97.2	95.7
avg Cat	88.4	**93.0**	91.9
avg overall	92.8	**95.2**	93.4

Table 4.12: Font style classification results for varying the minimum amount of text patch rows contained in a texture on synthetic and historical catalogue data sets with varying font size.

Training Data

In this section the influence of the amount and the composition of the training data is analysed. The purpose of this analysis is to provide general guidelines and highlight problems in terms of the selection of training data for the proposed font style classification approach.

First the influence of the amount of training data is evaluated in order determine the minimum amount training samples needed to achieve stable recognition rates. This is especially important since the training data is segmented and labelled manually. Which means that reducing the amount of training data also reduces the effort in terms of training data generation. For the evaluation of the amount of training data the classification results for data set SynStyle with a varying amount of training samples (25-1000) are computed. The gathered results are shown in the chart in Figure 4.5. The chart shows that the performance exceeds $RR = 90\%$ when using 75 or more training samples. Using 100 training samples a RR of 94.4% is achieved which is 2.2% higher compared to 75 samples. The maximum RR of 97.7% and is achieved using 900 training samples. This is 3.2% higher compared to 100 samples representing an average increase of 0.1% per 25 samples. Based on these results the minimum amount of training samples is set to 100. This value provides more stable results compared to 75 samples and reaches up to 95% of the maximum performance.

Another aspect that should be pointed out is the fact that the training, as well as the test,

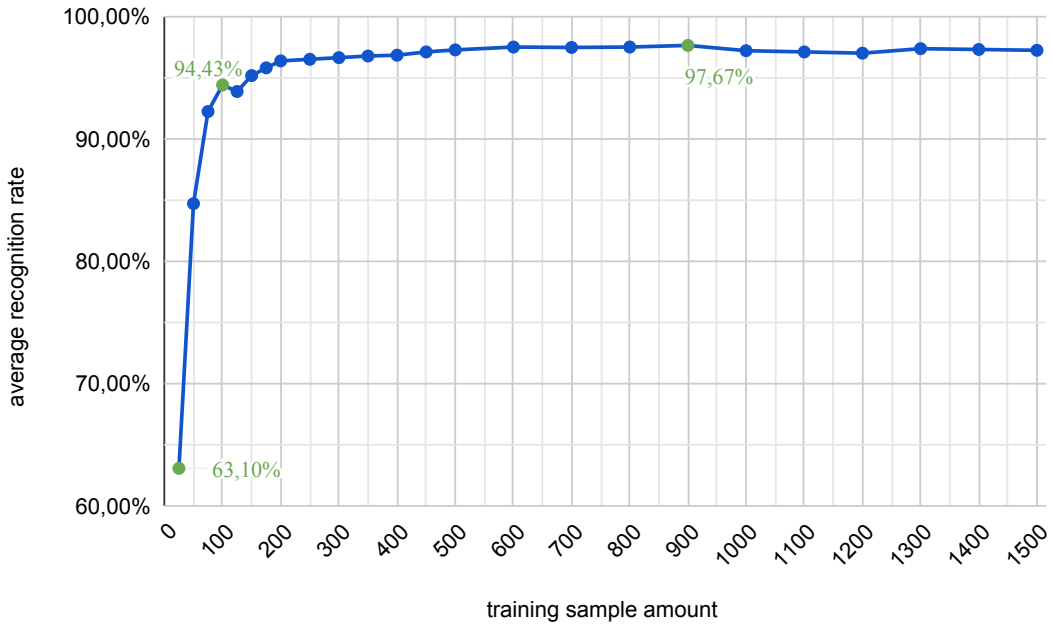


Figure 4.5: Chart showing the performance for a varying amount of training samples for the data set SynStyle.

data set contain duplicates. The reason for this is that the input regions are not analysed with respect to their text content. Therefore, the synthetic data set are generated from continuous text regions which might as well contain duplicates. However, the use of training data sets that do not contain duplicates can improve the recognition performance. For example, on data set SynStyle the use of a training data set containing 100 unique samples per class achieves a RR of 95.6%. This is an improvement of 1.2% over the training data set including duplicates. However, training data sets containing duplicates replicate the catalogue training data more closely and are therefore used by default.

Finally, the influence of the word length on the recognition performance is evaluated. As mentioned by Ma and Doermann [MD03] a single word might not provide enough texture information for a successful classification. In order to determine how the word length (number of characters) influences the overall classification performance detailed results are computed for words of varying length.

Based on the font styles and training data of data set SynStyle a set of 4469 unique samples (words) per class are analysed. The classification results for these samples are then split into groups according to the length of the words. The gathered results are shown in the chart in Figure 4.6. Based on the chart it can be said that the recognition performance is correlated with the word length. The highest performance with a RR of 96.3% over all classes corresponds to the group containing only words with more than

10 characters. The group containing only words with one or two characters yields the worst performance with 67.9%. With an increasing number of characters the average recognition rate increases as well. For words with five or more characters the *RR* is above 90%.

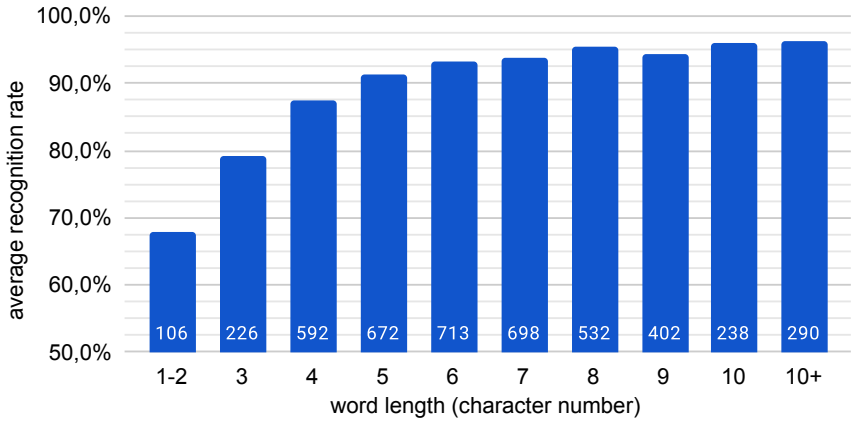


Figure 4.6: Chart showing the performance for test sets containing words of varying length.

Based on the word length evaluation it is concluded that short words (i.e. less than five characters) are especially challenging in terms of the proposed classification problem. Ma and Doermann [MD03] suggest using a special class for challenging words that provide too few texture information for a successful classification. Based on this suggestion multiple classes or an additional classifier could be used to deal with short words. In the proposed classification method only one classifier is used by default. However, dependent on the actual application the use of an additional classifier for short words might be used to improve the results.

4.2.5 Cropped Word Recognition on Historical Catalogues

The performance of the font style classification on the historical catalogues is evaluated based on the results on the data sets Cat1-3 (see Section 4.2.1 for details). In the data sets Cat1-2 four classes (font styles) and in the data set Cat3 three classes representing words of varying logical roles are contained. The classes of the data sets represent common logical roles (i.e. artist, address) but their font styles vary for each data set. In order to provide an overview of the classification results a confusion matrix that shows the joint performance for each class (i.e. logical role) over all three data sets is shown in Table 4.13.

The rows of the confusion matrix represent the predicted classes and the columns represent the actual, ground truth, classes. The values in the table represent the ratio between the predictions and the actual samples in percent for each pair of classes. This means that

		ground truth				
		artist	address	ID	title	#
predicted	artist	**99.4**	0.3	1.8	1.4	192
	address	0.0	**93.8**	8.5	3.2	407
	ID	0.0	1.0	**85.7**	2.1	211
	title	0.5	4.9	4.0	**93.4**	706
	#	178	389	224	725	1516

Table 4.13: Confusion matrix for classification results over all three catalogue data sets. The values represent the *RR* for each class in percent and the number of samples.

the i^{th} diagonal element represents the fraction of correct classifications (*True Positives*) for class i, i.e. the RR_{c_i} (see Equation 4.1), in percent. The *False Positives* are contained within the row and the *False Negatives* within the column that is aligned with the diagonal element. Additionally, the matrix also contains the number of samples/predictions (see last row/column #) for each class and the overall number of samples.

The joint *RR* in the confusion matrix shows that the class *artist* with 99.4% has the best and *ID* with 85.7% the worst performance. The lower performance on the class *ID* is caused by several factors. For *ID* a lower amount of training samples (see Figure 4.2.2) is used compared to the other classes. In addition, *ID* samples represent numerical identification numbers with up to three digits. This means that all samples of the class *ID* represent short words with less than five characters. As shown in Section 4.2.4 short words are especially challenging and therefore reduce the recognition performance. Presumably, the short length of the samples in the class *ID* are the main reason for its comparatively low *RR*. In fact, an in-depth analysis of the classification errors shows that about 78% of the classification errors over all classes and data sets occur on short words.

The confusion between the classes *ID* and *address* is explained by their content. *ID* contains only numerals and *address* also contains several numerals. Whereas the other two classes typically contain only literals. The second biggest confusion occurs between the classes *address* and *title*. The reason for this is that in data set Cat1 the font style of both classes only differs in terms of the font size.

In Table 4.14 the RR_{c_i} of each class and the *RR* for each catalogue data set are shown. These detailed results show that Cat1 is the most challenging data set ($RR = 90.1\%$) while Cat3 with only three classes yields the best performance ($RR = 97.2\%$). The difference in the performance is not only caused by the number of classes contained in the data sets but also by the visual discriminability of the contained font styles and the amount of image artefacts.

The results in Table 4.14 also show that it is possible to discriminate numerals from literals. The classes *ID* containing only numerals and the class *title* containing only literals in the data set Cat1 are both formatted with the same font style. Nevertheless,

	artist	address	ID	title	RR
Cat1	100.0	91.9	78.6	90.0	90.1
Cat2	100.0	96.4	76.0	95.0	91.8
Cat3	98.2	-	97.8	95.5	97.2

Table 4.14: Detailed recognition results for the historic catalogue data sets. The values are in percent and represent the *RR* for each class and the average *RR* over all classes.

the classes have a *RR* of 78.6% and 90.1%, respectively. Therefore, it is concluded that numerals and literals can be discriminated based on their texture features.

Similarly, the classes *artist* and *address* in Cat2 share the same font style but *artist* uses only capitalized letters. The corresponding classes have *RR*s of 100.0% and 96.4%, respectively. This means that also capitalized font styles can be discriminated based on their texture features.

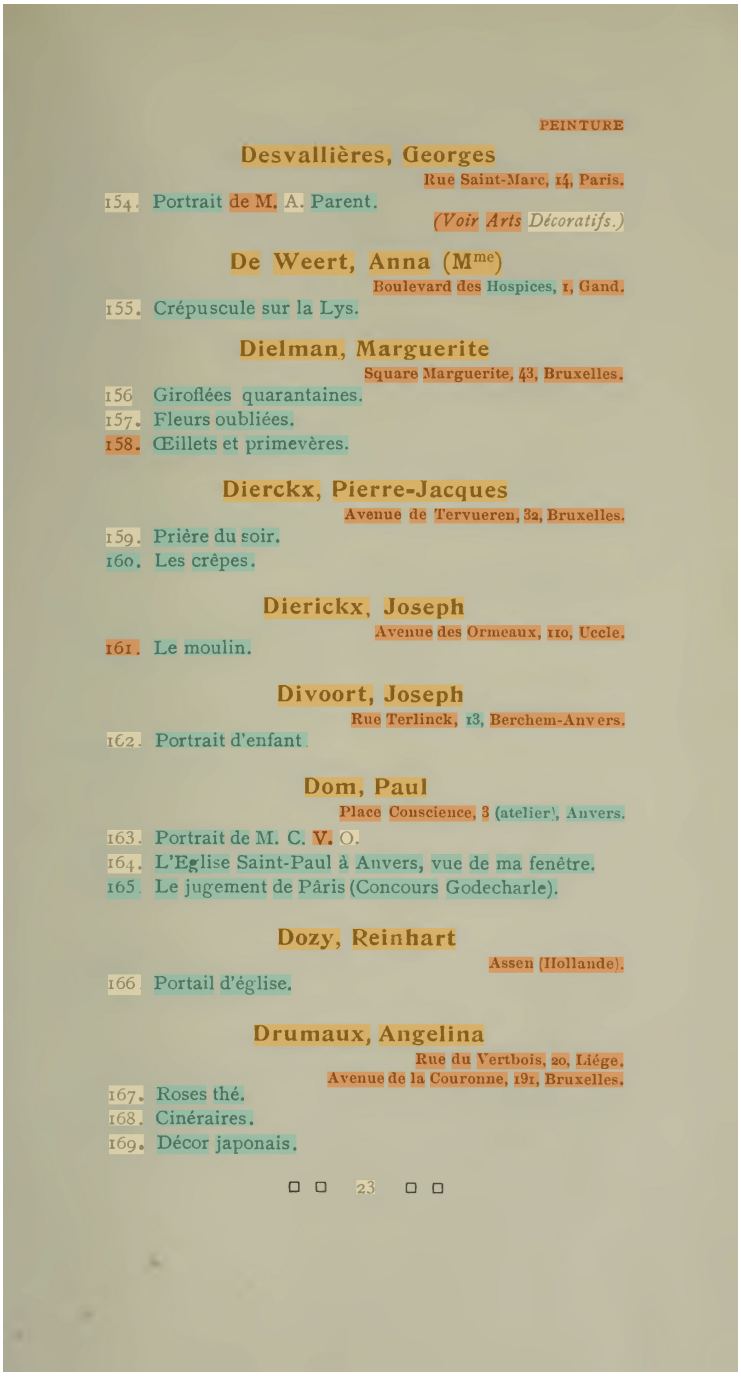
On the other hand, in data set Cat3 the class *artist* contains words representing first and second name. Both names share the same font style except that the second name is written in bold and the first name in regular style. Nevertheless, both names are classified as *artist* for simplicity reasons. The class trained with samples of first and second name yields a *RR* of 98.2%. This shows that similar font styles can be grouped together for training and used to classify similar fonts into a single class.

4.3 Historical Catalogue Processing

In this Section a brief evaluation of the combination of the page segmentation and the font style classification is presented. OCR results obtained by applying the Tesseract module to the catalogue images are not evaluated since the focus of this work is on the layout analysis and how it can be used to enhance the results of Tesseract.

The application of the font style classification to the text lines extracted by the page segmentation results requires the additional segmentation of word regions. In the final recognition system this segmentation is obtained by using the integrated Tesseract engine. In order to provide an analysis that is not influenced by errors in the word segmentation an evaluation based on ground truth word regions is done. This represents a ceiling analysis for the combination of the layout analysis stages. For this experiment ground truth word regions are mapped to the page segmentation results and then font style classification results are computed. An image illustrating the classification results for a page from the data set Cat1 is shown in Figure 4.7.

The figure shows that all regions are assigned to one of four pre-trained font style classes. The assigned font style labels are indicated by colouring the bounding box of the word regions with the respective label colour. The figure shows that the majority of the false classifications occur on short words but there are also false classifications of longer words



label colors: artist address ID title

Figure 4.7: Image illustrating the font style classification results for a page from the data set Cat1 [Arcc]. The results are computed based on the text line regions extracted by the page segmentation stage and an additional projection of the ground truth word regions.

		ground truth				
		artist	address	ID	title	#
predicted	artist	**96.1**	0.0	5.9	0.6	188
	address	1.7	**94.3**	11.7	3.9	424
	ID	0.6	1.0	**74.3**	2.1	185
	title	1.7	4.6	8.1	**93.5**	715
	#	178	389	222	723	1512

Table 4.15: Confusion matrix for classification results over all three catalogue data sets obtained by the combination with the page segmentation results. The values represent the *RR* for each class in percent and the number of samples.

especially for the class *address*. These errors are caused by the confusion with the class *title* that can only be discriminated due to its slightly smaller font size.

In Table 4.15 the summarized font style classification results based on the page segmentation regions are shown in form of a confusion matrix. The matrix contains the combined results from the same three catalogues and their selected pages that are also evaluated in the section on cropped word segmentation in Table 4.13. The only difference is that the word regions are obtained by mapping the ground truth regions to the segmented text lines rather than directly using tightly cropped ground truth word regions.

The results of the experiment show the performance on the extracted text line regions provides similar results to the cropped word results with one obvious deviation. The *RR* of the class representing the identification numbers yields a clearly lower performance with 74.3% compared to 85.7% on the cropped words images. Therefore, it is concluded that the classification performance of the class ID is negatively influenced by the text line segmentation. The class ID represents small regions which makes it vulnerable even to small segmentation errors as the ones shown in Section 4.1.3.

Considering the overall results, it can be said that the proposed combination of the layout analysis stages seems to be suitable for the use in the recognition system. However, the results also show that the performance issues of the font style classification on short words are intensified by errors in the word segmentation.

CHAPTER 5

Conclusion

In this thesis an approach for the extraction of structural and semantic information from historical art exhibition catalogues is proposed. The approach can be combined with the freely available state-of-the-art OCR engine Tesseract in order to ease the process of automating the extraction of specific information from catalogues. The additional information extraction consists of a two-stage process. First a page segmentation approach based on the extraction of MSER and a text region grouping stage are applied. This approach is based on state-of-the-art methods in page segmentation and provides optional functionality for white space analysis, black separator detection and marginalia segmentation. In the second stage the words contained in the segmented text line regions are further classified according to their font style. For this purpose, a predefined set of font styles is trained and input regions are classified by using Gabor filtering. The obtained font style classification results are then used to discriminate text regions representing specific categories of information such as identification number, artist name, etc. in the catalogue images. Finally, directions for the integration of Tesseracts [tes] OCR methodology into the system are given in order to provide text recognition results for the classified text regions. The individual stages of the system are discussed in more detail subsequently.

The page segmentation methodology of the proposed system is evaluated on the HBR2013 data set. The results show that adopting a robust text region grouping process adapted to historical documents is able to outperform Tesseract in terms of segmentation. In addition, the performance with respect to the historical books evaluated in the competition is improved by applying additional segmentation refinements (i.e. white space analysis, marginalia segmentation, etc.). However, the method lacks a non-text filtering stage in order to process images containing graphical content. Considering the segmentation of the catalogues, that consist of foremost text only pages, only the initial text region grouping methodology is used. This adaptation accounts for the sparse layout of the catalogues that typically contains several gaps and indents. Therefore, compared to the generic

page segmentation of Tesseract this approach reduces the issue of oversegmentation on the catalogues. However, the adaptation of the segmentation approach to other types of documents requires expert knowledge. This means that the approach provides specific abilities to deal with historical document images but lacks in terms of abilities to automatically adapt to varying types of documents.

The font style classification is evaluated using synthetic and catalogue images. The evaluation shows that adapting the parameters of the Gabor filters to the task of font style classification provides an improvement compared to standard parameter settings. Furthermore, it is shown that the method can be used to discriminate text regions based on various font characteristics (weight, typeface, size, etc.) and their content (i.e. numerals from literals). The proposed method is robust to noise and minor segmentation errors providing an advantage over approaches based on typographical features. Furthermore, the method is able to provide word-level classification results with recognition rates above 90% for cropped word images. However, the drawbacks of the method are its dependence on the word length and the requirement of training data. Therefore, manual effort is still required to acquire training data and correct classification errors.

Finally, the recognition system combines the page segmentation and font style classification results with the OCR methodology of Tesseract [tes]. The result of the system is a list of categorized text fragments that can be used for further research. The additional categorization and improved segmentation of text regions in the proposed system represents an advantage compared to using solely Tesseract. This means that the proposed recognition system is able to ease the process of digitizing specific information contained in the historical art exhibition catalogues. Therefore, the answer to the research question is: Yes, it is possible to outperform generic state-of-the-art system in terms of the extraction of specific information by utilizing formatting characteristics of the catalogues.

The performance improvement is achieved by adapting the individual parts of the system to the specific requirements of the catalogues and utilizing knowledge about their formatting. The process of integrating formatting characteristics can also be applied to other types of documents. However, the adaptation of the page segmentation to specific document types usually involves expert knowledge. Therefore, improving the adaptability of the page segmentation stage makes it possible to apply the proposed system to a wider range of document types.

5.1 Future Work

In order to overcome some of the limitations of the individual stages of the proposed system specific improvements can be applied to further refine the results. The performance of the page segmentation can be improved by applying a more elaborate pruning strategy and an additional non-text filtering stage [WFS+15]. For performance improvements in the font style classification other classifier types such as a Support-Vector Machine (SVM) and additional use of synthetic training data can be applied.

However, in order to improve the final results of the system the combination of the stages needs to be refined. This can be done by combining the geometrical features extracted in the page segmentation with the font style results in an additional classification stage. This additional classifier can be used to incorporate local geometrical features (e.g. height, absolute and relative position of text regions) in order to correct erroneous or insufficient font style information.

In general, the workflow of the proposed system can be improved by incorporating the stages of geometrical, logical and textual information extraction closer. In the proposed system these individual stages are executed in isolation and only their final results are combined by mapping them to each other. This can be improved by incorporating the texture-based analysis of text regions earlier in the page segmentation process. This could be used for the purpose of non-text filtering and in order to exploit similarity properties for grouping homogeneous text regions. The effectiveness of incorporating an efficient non-text classification into the page segmentation is demonstrated by the method using a minimum homogeneity algorithm by Tran et al. [TON+17, TNK16].

Furthermore, the extraction of formatting characteristics can be done by using a clustering approach. As proposed by Mehri et al. [MGKH+13] texture features can be used to segment historical documents into clusters of homogeneous regions. Where the clusters represent text regions of distinct font styles. The advantage of applying this kind of approach is that only the number of clusters needs to be determined in advance. Therefore, the need for training specific font styles and the corresponding training data generation is eliminated.

Acronyms

BB Bounding Box. 13, 14, 28–30, 33, 39, 40, 44, 45, 53, 54, 61, 62

BNN Below Nearest Neighbour. 42, 43

CC Connected Component. 11–16, 22, 26

DIA Document Image Analysis. 4, 6, 7, 19

DIBCO Document Image Binarization Competition. 9

ER Extremal Region. 26

GLCM Grey Level Co-occurrence Matrix. 18, 46

GMM Gaussian Mixture Model. 17

HBR2013 ICDAR2013 Competition on Historical Book Recognition. 3, 5, 6, 10, 15, 19, 23, 25, 27, 28, 31, 35, 38, 42, 44, 45, 53, 57–62, 83

HVS Human Visual System. 46

ICDAR International Conference on Document Analysis and Recognition. 4, 10–12, 14, 15, 19, 23

k-NN k-Nearest Neighbours. 13, 52, 66

LSD Line Segment Detector. 29, 40

MHA Minimum Homogeneity Algorithm. 15

MSER Maximally Stable Extremal Region. 9, 10, 23–30, 35, 37, 59, 83

MST Minimum Spanning Tree. 13

NACF Normalized Autocorrelation Function. 24

NN Neural Networks. 14

NN_r Right Nearest Neighbour. 30–33, 35

OCR Optical Character Recognition. 3, 4, 6, 7, 15–17, 19, 21, 22, 55, 57, 59, 79, 83, 84

OFR Optical Font Recognition. 7, 8, 16, 17

PMF Probability Mass Function. 24, 25

RDCL Competition on Recognition of Documents with Complex Layouts. 15, 19

RLSA Run-Length Smearing Algorithm. 11–13

RXYC Recursive X-Y Cuts. 11

SIFT Scale Invariant Feature Transform. 9, 10

SVM Support-Vector Machine. 84

WED Weighted Euclidean Distance. 51, 52, 66

Bibliography

[ACPP11] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. Historical Document Layout Analysis Competition. In *International Conference on Document Analysis and Recognition*, pages 1516–1520, 2011.

[ACPP13a] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. ICDAR 2013 Competition on Historical Book Recognition (HBR 2013). In *International Conference on Document Analysis and Recognition*, pages 1459–1463, 2013.

[ACPP13b] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. ICDAR 2013 Competition on Historical Newspaper Layout Analysis (HNLA 2013). In *International Conference on Document Analysis and Recognition*, pages 1454–1458, 2013.

[ACPP15] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. ICDAR2015 Competition on Recognition of Documents with Complex Layouts - RDCL2015. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1151–1155, 2015.

[AF00] A Amin and S Fischer. A Document Skew Detection Method Using the Hough Transform. *Formal Pattern Analysis & Applications*, 3:243–253, 09 2000.

[ale] Aletheia - Document Analysis System. `https://www.primaresearch.org/tools/Aletheia`. last accessed on: 26-06-2020.

[Ant98] Apostolos Antonacopoulos. Page Segmentation Using the Description of the Background. *Computer Vision and Image Understanding*, 70(3):350–369, 1998.

[Arca] The Internet Archive. Esposizione Internationale d'Arte - Venezia 1905. `https://archive.org/details/catalogo6190bien`. last accessed on: 07-07-2020.

[Arcb] The Internet Archive. Salon d'Automne - Paris 1907. `https://archive.org/details/cataloguedesouvr00salo`. last accessed on: 07-07-2020.

[Arcc] The Internet Archive. Salon des Beaux-Arts - Bruxelles 1907. https://archive.org/details/expositiongn1907expo. last accessed on: 07-07-2020.

[art] Research project: Exhibitions of Modern European Painting 1905-1915 (funded by the Austrian Science Fund FWF, Project Number P 29997-G24). https://exhibitions.univie.ac.at/info/project. last accessed on: 26-06-2020.

[Bai92] Henry S. Baird. Background Structure In Document Images. In *Advances in Structural and Syntactic Pattern Recognition*, pages 17–34. World Scientific, 1992.

[Bai95] Henry S. Baird. Document Image Analysis. chapter The Skew Angle of Printed Documents, pages 204–208. IEEE Computer Society Press, 1995.

[Bre02] Thomas M. Breuel. Two geometric algorithms for layout analysis. In *International workshop on document analysis systems*, pages 188–199, 2002.

[BT14] Henry S. Baird and Karl Tombre. The Evolution of Document Image Analysis. In David Doermann and Karl Tombre, editors, *Handbook of Document Image Processing and Recognition*, pages 63–71. Springer London, London, 2014.

[CAP17] C. Clausner, A. Antonacopoulos, and S. Pletschacher. ICDAR2017 Competition on Recognition of Documents with Complex Layouts - RDCL2017. In *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1404–1410, Nov 2017.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[CYL13] K. Chen, F. Yin, and C. L. Liu. Hybrid Page Segmentation with Efficient Whitespace Rectangles Extraction and Grouping. In *12th International Conference on Document Analysis and Recognition*, pages 958–962, 2013.

[DT14] David Doermann and Karl Tombre. Text Recognition. In David Doermann and Karl Tombre, editors, *Handbook of Document Image Processing and Recognition*, pages 255–486. Springer London, London, 2014.

[EGKO17] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. A comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition*, 64:1–14, 2017.

[Fre] ABBYY FineReader. https://www.abbyy.com/de-de/finereader/. last accessed on: 26-06-2020.

[Fuk90] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition (2nd Ed.)*. Academic Press Professional, Inc., New York, 1990.

[GSD11] A. Garz, R. Sablatnig, and M. Diem. Layout Analysis for Historical Manuscripts Using Sift Features. In *2011 International Conference on Document Analysis and Recognition*, pages 508–512, Sep. 2011.

[HSD73] Robert M. Haralick, K. Sam Shanmugam, and Its'hak Dinstein. Textural Features for Image Classification. *IEEE Trans. Systems, Man, and Cybernetics*, 3:610–621, 1973.

[JB92] Anil K. Jain and Sushil Bhattacharjee. Text Segmentation Using Gabor Filters for Automatic Document Processing. *Machine Vision and Applications*, 5(3):169–184, July 1992.

[JZ96] Anil K. Jain and Yu Zhong. Page segmentation using texture analysis. *Pattern Recognition*, 29(5):743 – 770, 1996.

[Kis14] Koichi Kise. Page Segmentation Techniques in Document Analysis. In *Handbook of Document Image Processing and Recognition*, pages 135–175. Springer London, 2014.

[KNSV93] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):737–747, July 1993.

[KSI98] Koichi Kise, Akinori Sato, and Motoi Iwata. Segmentation of Page Images Using the Area Voronoi Diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[MCUP04] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[MD03] Huanfeng Ma and David Doermann. Gabor Filter Based Multi-class Classifier for Scanned Document Images. In *7th International Conference on Document Analysis and Recognition*, pages 968–972, Aug 2003.

[MD04] Huanfeng Ma and D. Doermann. Adaptive word style classification using a Gaussian mixture model. In *Proceedings of the 17th International Conference on Pattern Recognition ICPR04*, volume 2, pages 606–609, Aug 2004.

[MGKH+13] Maroua Mehri, Petra Gomez-Krämer, Pierre Héroux, Alain Boucher, and Rémy Mullot. Texture Feature Evaluation for Segmentation of Historical Document Images. pages 102–109, 08 2013.

[MGKHM13] Maroua Mehri, Petra Gomez-Krämer, Pierre Héroux, and Rémy Mullot. Old document image segmentation using the autocorrelation function and multiresolution analysis. volume 8658, 02 2013.

[NJ07] A. Namboodiri and A. Jain. Document Structure and Layout Analysis. *Digital Document Processing*, pages 29–48, 2007.

[NM12] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3538–3545, June 2012.

[NS84] G. Nagy and S. C. Seth. Hierarchical image representation with application to optically scanned documents. In *7th International Conference on Pattern Recognition (ICPR)*, pages 347–349, 1984.

[O'G93] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, Nov 1993.

[PD14] Umapada Pal and Niladri Sekhar Dash. Language, Script, and Font Recognition. In David Doermann and Karl Tombre, editors, *Handbook of Document Image Processing and Recognition*, pages 291–330. Springer London, London, 2014.

[PYR13] R. Pintus, Y. Yang, and H. Rushmeier. ATHENA: Automatic text height extraction for the analysis of old handwritten manuscripts. In *2013 Digital Heritage International Congress (DigitalHeritage)*, volume 1, pages 605–612, Oct 2013.

[PZ91] T. Pavlidis and J Zhou. Page Segmentation by White Streams. In *First International Conference on Document Analysis and Recognition*, pages 945–953, 1991.

[PZKG18] Ioannis Pratikakis, Konstantinos Zagoris, Panagiotis Kaddas, and Basilis Gatos. ICFHR 2018 Competition on Handwritten Document Image Binarization (H-DIBCO 2018). pages 489–493, 08 2018.

[QM16] S. Qin and R Manduchi. A fast and robust text spotter. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8, March 2016.

[RDB07] Jean-Yves Ramel, Marie-Luce Demonet, and Débastien Busson. User-driven Page Layout Analysis of historical printed Books. *International Journal On Document Analysis and Recognition*, 9:243–261, 04 2007.

[SKH+13] Fouad Slimane, Slim Kanoun, Jean Hennebert, Adel M. Alimi, and Rolf Ingold. A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution. *Pattern Recognition Letters*, 34(2):209–218, 2013.

[Smi09] Ray Smith. Hybrid Page Layout Analysis via Tab-Stop Detection. In *10th International Conference on Document Analysis and Recognition*, pages 241–245, 2009.

[SPJ97] A. Simon, J.-C. Pret, and A. P. Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):273–277, March 1997.

[Tan92] T. N. Tan. Texture feature extraction via visual cortical channel modelling. In *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol. III. Conference C: Image, Speech and Signal Analysis,*, pages 607–610, Aug 1992.

[tes] Tesseract Open Source OCR Engine. `https://github.com/tesseract-ocr`. last accessed on: 26-06-2020.

[TNK16] Tuan Anh Tran, In Seop Na, and Soo Hyung Kim. Page Segmentation Using Minimum Homogeneity Algorithm and Adaptive Mathematical Morphology. *International Journal on Document Analysis and Recognition*, 19(3):191–209, September 2016.

[TON+17] Tuan Anh Tran, Kanghan Oh, In-Seop Na, Guee-Sang Lee, Hyung-Jeong Yang, and Soo-Hyung Kim. A robust system for document layout analysis using multilevel homogeneity structure. *Expert Systems with Applications*, 85:99–113, 2017.

[VGJMR10] R Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[WCW82] K. Y. Wong, R. G. Casey, and F. M. Wahl. Document Analysis System. *IBM Journal of Research and Development*, 26(6):647–656, 1982.

[WFS+15] L. Wang, W. Fan, J. Sun, S. Naoi, and T. Hiroshi. Text line extraction in document images. In *13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 191–195, Aug 2015.

[XY03] Yi Xiao and Hong Yan. Text region extraction in a document image based on the Delaunay tessellation. *Pattern Recognition*, 36(3):799 – 809, 2003.

[ZI93] Abdel Wahab Zramdini and Rolf Ingold. Optical Font Recognition from Projection Profiles. *Electronic Publishing*, 6:249–260, 1993.

[ZI98] Abdelwahab Zramdini and Rolf Ingold. Optical font recognition using typographical features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:877–882, 1998.

[ZTW01] Yong Zhu, Tieniu Tan, and Yunhong Wang. Font recognition based on global texture analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 23(10):1192–1200, 2001.