FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Text Classification and Layout Analysis for Document Reassembling

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

Markus Diem
Registration Number 0226595

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

The dissertation has been reviewed by:

(Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig)	(Basilis G. Gatos, PhD)

Wien, March 10, 2014

(Markus Diem)

Erklärung zur Verfassung der Arbeit

Markus Diem
Mollardgasse 22/19, 1060 Wien

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.

(Ort, Datum) (Unterschrift Verfasser)

Danksagung

Ich möchte mich an dieser Stelle bei all jenen Menschen bedanken, die mich begleitet und unterstützt haben. Sie haben wesentlich dazu beigetragen, dass ich diese Arbeit schreiben konnte.

Für die fachliche Unterstützung möchte ich mich bei allen Arbeitskollegen am CVL bedanken, die in vielen Diskussionen meine Idee der Computer Vision geformt haben. Speziell möchte ich mich bei Melanie Gau, Michael Hödlmoser, Martin Kampel, Rainer Planinc, Michael Reiter, Sebastian Zambanini und Andreas Zweng bedanken. Ein Dank gilt auch Stefan, der mich nie zu seiner Masterparty eingeladen hat, Florian, dessen zum Teil absurde Ideen den Büroalltag auffrischen und Fabian, der mich schon beim Studium für seine nicht immer absurden Ideen begeisterte. Florian hat mich zusätzlich durch alle Stasi Projekte begleitet und mir dabei die Mathematik sowie eine gewisse Stressresistenz näher gebracht. Die letzteren drei hatten einen wesentlichen Einfluss auf die Entstehung des *Montagsbier*, welches in jeder gesunden Büroumgebung empfehlenswert ist.

Ein spezieller Dank gilt auch meinem Betreuer Robert Sablatnig. Er hat eine Büroatmosphäre geschaffen, die sowohl produktiv als auch belebend ist. Des Weiteren hat er mein Interesse an der Forschung geweckt und mich auf die Verfassung wissenschaftlicher Artikel vorbereitet. Begriffe wie *many, more* und jeglicher Konjunktiv verwendete ich weit häufiger ohne seine Korrekturen. Bei Basilis Gatos möchte ich mich für die Zeit bedanken, die er in das Review dieser Arbeit investiert. Seine wissenschaftliche Arbeit ist die Grundlage einiger Entwicklungen die hier vorgestellt werden.

Ein abschließender Dank gilt meiner Familie, die mich zu jenem Menschen formte, der ich nun bin. Meine Großmutter Frieda förderte besonders in meiner Kindheit meinen Wissensdurst. Lukas hat immer Zeit für Diskussionen über Gott, die Welt und im speziellen die Informatik. Im Gegensatz dazu holt mich Clemens aus dieser Welt gerne heraus und zeigt mir jene, in der die Ästhetik der Form dominiert. Meine beiden Schwestern Sarah und Sabine holen mich wiederum gerne auf den Boden der Realität. Ein letzter Dank gilt Alex, Sandro, Lara, Nina, Werner und Annemarie, die immer da sind wenn es notwendig ist.

Diese Dissertation widme ich meinen Eltern und Babsi. Meine Eltern haben mein Studium ermöglicht und mir für meinen Lebensweg Werte mitgegeben, die sich bis heute als sehr gut erwiesen haben. Babsi, du bist nach Platons Symposion meine fehlende Hälfte.

Danke

“as goht als”

Abstract

In the context of automated reassembling of manually torn document snippets contour based approaches are insufficient because snippets have the same rupture edges if more than one page is torn at the same time. Moreover jigsaw puzzling is NP hard which requests for a grouping of document snippets beforehand such that the complexity and computational speed of reassembling is improved. Analyzing the visual content of document snippets renders the distinction of snippets with the same contours possible. In addition, a visual content extraction enables fine alignment of snippets with the same content and for grouping snippets. The document analysis approaches presented in this thesis are part of a combined reassembling which utilizes content and contour for the reconstruction of about 600 Million Stasi snippets.

The ruling analysis classifies the supporting material into *void*, *lined*, and *checked* paper. If a ruling is detected, the lines are localized accurately which allows for snippet alignments. Snippets might have sparse visual content depending on the conscientiousness when tearing. Therefore a new word localization – the so-called Profile Box – is introduced which keeps a compact word representation while accounting for anticipated deformations such as a word's local skew. These word boxes are further classified into *printed*, *manuscript*, and *non-text* elements by means of Gradient Shape Features (GSF) which are designed newly for this task. The latter class allows for rejecting falsely binarized elements which improves the robustness in the presence of degraded or noisy documents. Finally, a layout analysis is performed that is based on a bottom-up approach to keep the element clustering flexible even if a global text structure is not present. Results on various publicly available databases show that the methodology is capable of being adopted to different document analysis scenarios.

A synthetic database for ruling line removal is created and made publicly available which allows comparisons between the approach proposed and other state-of-the-art methodologies. The text classification is compared to other approaches by means of the PRImA benchmarking database and the IAM database, which is a handwriting database written by multiple authors. The methodology presented achieves the best results in both empirical evaluations. On real world Stasi snippets, the recognition rate is lower because of the heterogeneity and sparseness of content in the data. The layout analysis is additionally evaluated on the most recent Handwriting Segmentation Contests where it competes state-of-the-art methods and on a medieval database.

Kurzfassung

Eine automatische Dokumentrekonstruktion von handzerrissenen Akten ermöglicht die Wiederherstellung von verlorengeglaubtem Inhalt. Das zugrundeliegende Datenmaterial beinhaltet 600 Millionen Stasi Schnipsel die beim Fall der Berliner Mauer vernichtet wurden. Konturbasierte Ansätze können Schnipsel nicht richtig zusammensetzten wenn mehrere Seiten gleichzeitig zerrissen wurden. Zusätzlich ist die Komplexität der automatischen Rekonstruktion zu hoch wenn jedes Schnipsel mit jedem verglichen werden muss. Deshalb wird vor der Rekonstruktion der visuelle Inhalt von Schnipseln analysiert. Dadurch kann einerseits eine Vorsortierung vorgenommen werden, andererseits ermöglicht es Schnipsel mit gleichen Risskanten voneinander zu unterscheiden. Algorithmen zur automatischen Textlokalisierung und Papieranalyse werden in dieser Doktorarbeit vorgestellt, die bei der Rekonstruktion mit konturbasierten Ansätzen kombiniert werden.

Die Papieranalyse klassifiziert Papier in liniiert, kariert und leeres Papier. Liegt ein liniiertes oder kariertes Schnipsel vor, so werden die Linien genau lokalisiert um benachbarte Schnipsel basierend auf deren Liniierung auszurichten. Des Weiteren wurde eine neue Textlokalisierung entwickelt, die Wörter kompakt repräsentiert und deren lokale Ausrichtung genau wiedergibt. Alle Elemente eines Dokuments werden in Maschinenschrift, Handschrift oder kein Text mit Hilfe von sogenannten *Gradient Shape Features* (Gsf) klassifiziert. Die Erkennung von *kein Text* ermöglicht es falsch binarisierte Elemente zu verwerfen um auf diese Weise gute Ergebnisse in verrauschten Dokumenten zu erzielen. Nach der Klassifikation und Lokalisierung von Text werden diese Elemente zu hierarchisch höheren Strukturen zusammengefasst. Dabei wird ein bottom-up Verfahren verwendet welches auch auf zerrissenen Dokumenten angewendet werden kann.

Die Methodik wurde empirisch auf öffentlich verfügbaren Datensätzen evaluiert und mit bestehenden Dokumentanalysesystemen verglichen. Die Textklassifikation und Lokalisierung konnte dabei bisherige Ergebnisse auf einem Datensatz mit modernen gedruckten Layouts und einem Handschriftendatensatz verbessern. Die Layout Analyse wurde auf den letzten drei Page Segmentation Contest Datensätzen evaluiert. Dort wurden im Vergleich zu State-of-the-Art Methoden ähnliche Ergebnisse erzielt, wobei sich herausstellte, dass besonders Bangla eine Schwierigkeit für die entwickelte Methode darstellt.

Contents

CHAPTER 1

Introduction

Document analysis in the context of computer vision aims at automatically extracting information from digitized documents. In order to narrow the variety of data, *documents* are referred to 2D supporting material (e.g. paper) that contain text and/or graphical elements. Document analysis applications include amongst others ruling analysis [LK10], text classification [ZLD04; Kan+07], writer identification [FS13], page segmentation [KC12; Sta+13; Ant+09a], word spotting [Fri+12; Fis+12], cursive handwriting recognition [MB02], and OCR [BSM99]. Text recognition aims at making a document's textual content digitally accessible. The motivations of automated text classification and page segmentation are manifold including pre-processing for text recognition [KHK08], pre-processing for form extraction [PB11], image compression [ITW93], or document retrieval [Pen+09]. Similar to the motivation of Peng et al. [Pen+09], the document analysis tasks presented in this thesis aim at document clustering and retrieval. Although, the methods presented are an intermediate processing stage of document reassembling, all methods can be applied for general document analysis tasks too which is shown by means of empirical evaluations on publicly available databases.

The safety curtain which separated East Germany from West Germany dissolved in 1989 after the Fall of the Wall. The *Stasi*[1] who kept the citizens under surveillance tried to annihilate secret records in order to protect them from unauthorized access. Since people occupied their offices, they were not able to destroy all records [SN08]. Nowadays (2014), the records are partially made accessible. However, 600 Million[2] manually torn document snippets are secured. These snippets shall be made accessible digitally so that the history during the Cold War can be rehabilitated. Therefore software is developed which is capable of reassembling these snippets. This thesis deals with the visual content analysis of document snippets which is needed for reassembling document snippets.

Document clustering and reassembling based on the visual content present in doc-

[1] Ministerium für Staatssicherheit – Secret police in East Germany

[2] The Guardian, 10 May 2007 `http://www.theguardian.com/world/2007/may/10/germany.kateconnolly1`

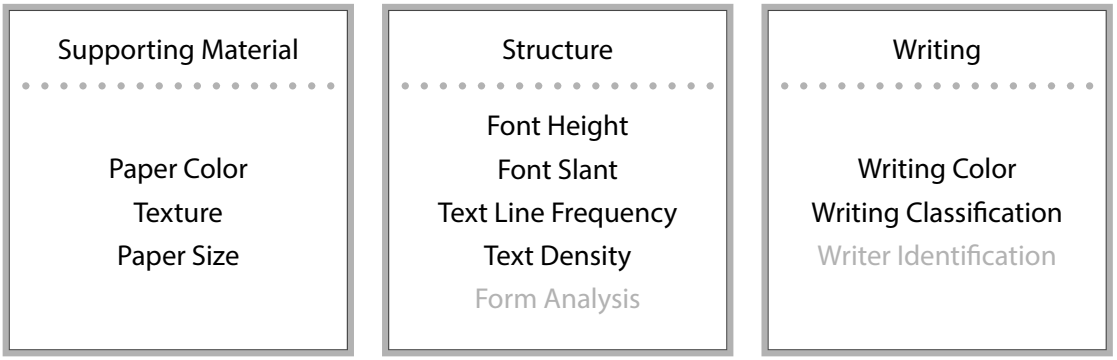


Figure 1.1: Features automatically extracted during document analysis for document clustering. Gray features are not discussed in this thesis.

uments is achieved by extracting features which are listed in Figure 1.1. Gray items including Form Analysis and Writer Identification are not further discussed in this thesis. The features are disposed into three groups. The first of which summarizes features that analyze a document's supporting material. Chapter 3 discusses the methodology used for extracting these features. The second group is called *structure* which contains features extracted from the layout and composition of text. The process of extracting these features robust with respect to heterogeneous and sparsely inscribed documents is discussed in Chapter 5. Features of the third and last group *writing* annotate text which is present in documents. Chapter 4 gives an outline of the methodology developed for text localization and classification which details the algorithms that proved to be best suited for the task at hand. All features extracted consider the composition of documents and allow therefore to create a fingerprint which is unique for each documents but allows for grouping similar documents. An overview of the methods presented in this thesis can be found in [KDS09; DKS09; DKS10; Kle+11; Die+14].

1.1 Motivation

Considering a mass of manually torn documents, a manual reassembling is time consuming and therefore expensive. If an automated reassembling is applied, challenges arise which are discussed subsequently. If more than one page is torn at the same time, the edge rupture is not unique. Thus, reassembling torn documents by means of the contour does not necessarily result in documents correctly reassembled. Furthermore, reassembling a jigsaw puzzle is NP-hard. Hence, there is a need for grouping possible jigsaw pieces before puzzling which reduces the search space.

These physical constraints yielded the analysis of the snippets' content. The idea is simple: an automated analysis of the document's content allows for grouping documents which have similar visual features. Furthermore, information such as text line location or ruling line location can be used for aligning snippets. Even though the automated content

annotation of torn documents assists and speeds-up the reassembling of these documents, the extraction itself comes up with new challenges. General document analysis tasks such as layout analysis, text classification or ruling analysis are frequently discussed in the literature (see Section 2). Applying state-of-the-art analysis methods on document snippets is in general not applicable because of their variety and sparse content. The snippets are a collection of secret records which were produced between 1950 and 1989. Therefore, the documents are composed by different typewriters, carbon copies, authors, and layouts. These constraints establish a need for robust document analysis algorithms which are able to handle sparse data.

Features which are retrieved from a document's content allow for applications which are enumerated subsequently:

Document Clustering: Simple clustering methods can be used to group documents which have a similar appearance. By these means archivists can narrow their search field if they are provided with a mass of unsorted documents.

Document Retrieval: The annotation of a document's visual appearance allows for retrieving documents with similar appearance. This is especially useful if one wants to find all documents of a specific layout (e.g. forms) within a database.

Document Reassembling: The visual feature extraction narrows the search space for automated document reassembling. This reduces the amount of document pieces and therefore the computational complexity of reassembling. In addition, visual features resolve ambiguities of documents with the same or similar edge rupture.

Context Retrieval: The layout analysis groups document elements into words, text lines and text blocks. These annotations are the basis for Ocr.

All applications discussed assist archivists when dealing with a mass of unsorted digital documents. Even though visual features are retrieved, the context of a document – the words written – are not extracted. Extracting the context would allow for full text search, document retrieval and clustering not just on the basis of their visual appearance but also with respect to their context. Hence, the question arises why a translation to machine text is not performed. There are a few considerations that need to be made before developing Ocr for the database at hand. First, considering the success of commercial Ocr and recognition rates of scientific systems reported at the turn of the millennium [BSM99], Ocr of clean machine printed Latin documents is not a research issue anymore. Secondly, state-of-the-art cursive handwriting recognition methods are not accurate enough for a reliable machine translation [Fis+12].

1.1.1 Scope of Discussion

The automated extraction of visual content in documents and document snippets is discussed in this thesis. The extraction targets on the one hand automated reassembling of

torn documents and on the other hand grouping of visually similar documents. Therefore, the document features computed include the analysis of a document's supporting material, analysis of text and layout. In order to gain this information typical preprocessing steps such as binarization, document de-skewing and de-noising are applied. These pre-processing steps are not further discussed. Furthermore, subsequent processing steps in the processing chain of automated document reassembling and clustering such as the reassembling itself are not part of this thesis. Hence, the visual information extraction of torn documents is the object of investigation. This process is split into three processing stages which are discussed in more detail subsequently.

Figure 1.2 illustrates the annotation of text areas in a typewritten document. For text area annotation words are initially found by means of Connected Component (CC) analysis (see Section 4.2.1). The magnifier on the left shows these annotations. Each word box is classified into *noise*, *printed*, or *manuscript* text. The *noise* label allows for rejecting non-text foreground elements which were detected during binarization. These elements typically include illustrations, bleed-through text or noise such as stains. Differentiating between handwritten (blue rectangles) and printed (yellow rectangles) enables a grouping of documents into handwritten and printed documents. This grouping can be used for the selection of algorithms specialized on either handwritten or printed text, clustering documents and recognizing layout types such as forms.

Having labeled words, they are grouped to text lines and text boxes. The text lines which are shown in the right magnifier of Figure 1.2 represent the next hierarchical level of text composition. Text lines allow for aligning document snippets, computing attributes such as line spacing or text alignment and serve as basis for reading. The text line clustering which is discussed in Section 5.1 collects the classification probabilities of words and applies a neighborhood voting which corrects errors such as the "*space*" in the left magnifier. The last hierarchy level represents text boxes which are found by grouping text lines of similar skew and spacing. Additional features such as text color, text slant or text height improve the characterization of documents.

In addition to the three text annotation hierarchies which are illustrated in Figure 1.3 (right magnifier), the supporting material of documents is analyzed. Therefore, the supporting material color is extracted from a background patch. In addition a ruling analysis is performed. This analysis which is detailed in Section 3.1 allows for discriminating *lined*, *checked*, and *void* supporting material. The document ruling is assumed to be global and therefore not to change within a single document. The ruling classification is performed on a patch which contains as few text as possible.

1.1.2 Objective

A computer vision system is presented which is capable of analyzing the visual content of digitized documents. Since we have to deal with 600 million snippets, processing time is crucial. That is why, the algorithms are designed to gain performance on both sides, computational speed and accuracy. The question is if it is possible to extract a document's visual content accurately so as to allow for automated document reconstruction. For this task new computer vision methodologies are designed which address the problem

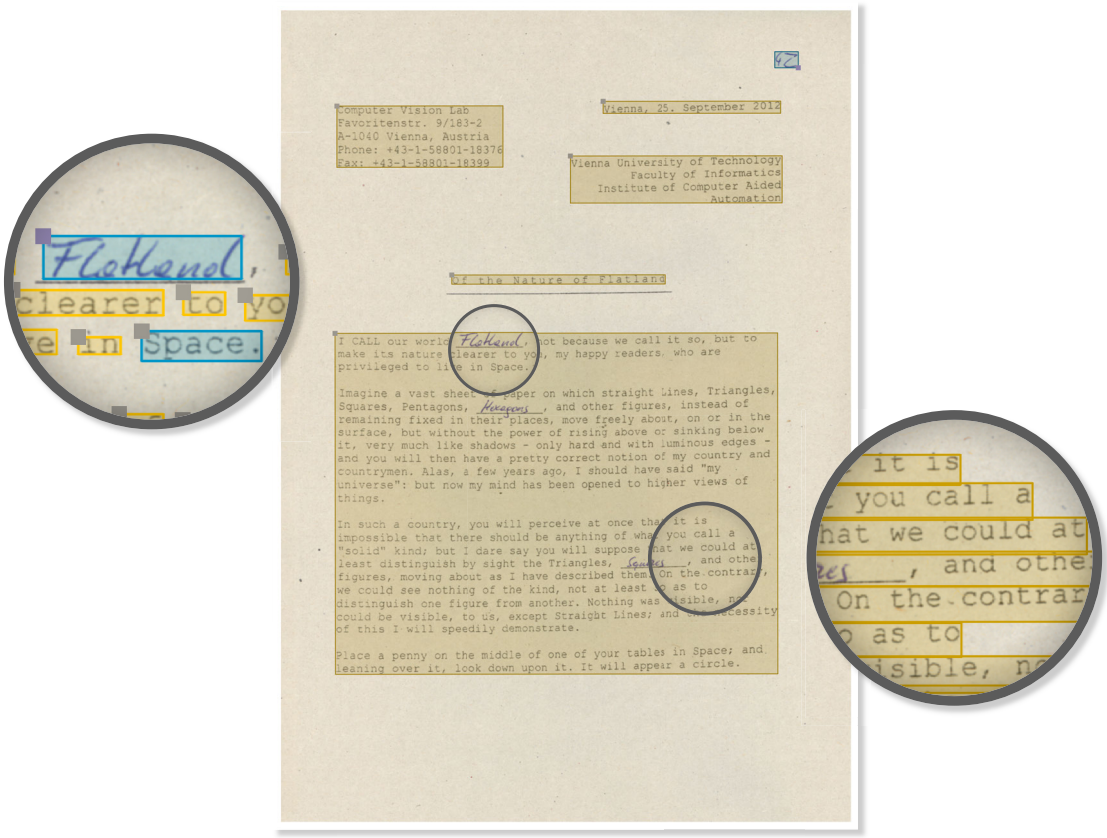


Figure 1.2: Word annotations (left magnifier) where blue labels indicate handwritten and yellow labels printed text. Text lines (right magnifier) and text blocks extracted automatically from a document image.

statement. The methodologies presented are approached by testing state-of-the-art document analysis methods empirically. Having analyzed the assets and drawbacks of these approaches, they are enhanced in order to increase their robustness. If the methodologies did not allow for an accurate information extraction, new algorithms are designed which compensate drawbacks of established ones.

The supporting material classification is to our knowledge a new approach of addressing ruling line analysis. Classifying supporting material prior to ruling line localization has advantages which are detailed subsequently. If the label (e.g. *checked*) of supporting material is known, the localization can be carried out in a sensitive way. Therefore, faded-out and spurious ruling lines are detected which cannot be found by means of binarization or line detection (e.g. Hough Transform). The system's capability of accurately classifying supporting material into *void*, *lined*, or *checked* is presented in Section 3.2. Alongside to the improvement of ruling line localization, the class label is used for document grouping.

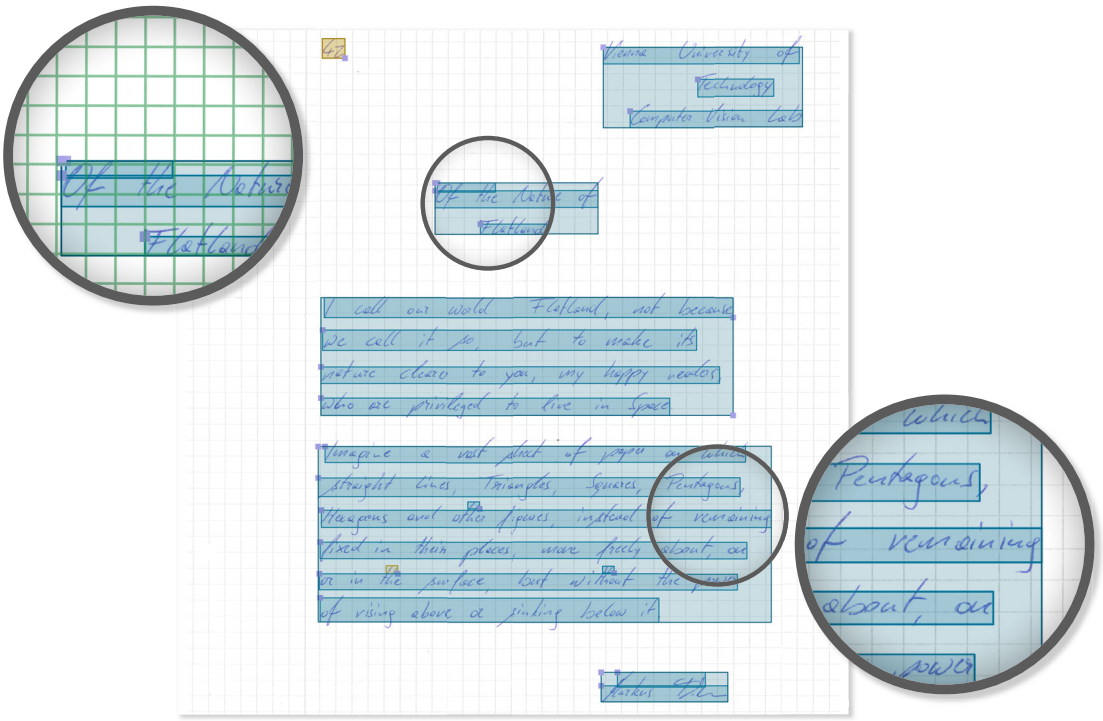


Figure 1.3: Layout analysis on a handwritten document page. Blue rectangles denote handwritten text lines and blocks. The left magnifier shows the ruling line classification and localization.

Binarization of degraded, multimodal, or ancient documents is an open research topic [PGN13]. Nevertheless, the binarization of documents has advantages that are outlined subsequently. It allows for a fast region of interest localization which is a simple CC analysis if binary data is retrieved. Assuming a perfect binarization, features for subsequent classification steps can be extracted which accurately characterize the elements while retaining a compact representation. These advantages are the reason for incorporating Damocles' sword binarization as the basis for text localization. Since binarization which does not incorporate an element wise classification cannot cushion all false positives a rejection of these elements is incorporated in the text classification. Therefore, an additional text class *noise* is trained which allows for detecting non-text elements and therefore to deal with binarization errors.

Although handwritten and printed text have different structures and topologies, a localization methodology is designed which is capable of dealing with both modalities. For a text classification into these two classes, the modalities are beneficial since they allow for automated labeling. Therefore a new feature – the Gradient Shape Feature (GSF) – is designed which can discriminate between these two types. They are fast to compute and – compared to other local descriptors – low dimensional which allows for fast classification.

Since document snippets may contain sparse up to no content at all, a need for flexible layout analysis is established. That is why, a bottom-up layout analysis is proposed which has no need for incorporating a-priori knowledge of a document's structure. Depending on the scenario, more complex grammars can be incorporated which allow for example to split two column text with low inter-column spacing. The layout analysis is carried out on rectangles which simplifies the representation of words or characters and therefore allows for a fast grouping of elements.

All methods presented give intermediate results in the context of a longer processing chain. Therefore, they are designed such that they do not just label elements but additionally provide accuracies which indicate the certainty of classification. The accuracies provided by the text classification and ruling analysis are further fed to a machine learning component which decides if a document snippet is a misfit or not.

1.2 Main Contribution

The research topic presented deals with document analysis. Therefore, the main contributions are relevant for document analysis and computer vision systems. Contributions include the evaluation of state-of-the-art methodologies on new databases, a new approach of recognizing the supporting material's structure, a new methodology of classifying text and a fast layout analysis which is capable of dealing with sparse text data.

Ruling Analysis: The classification of supporting material prior to ruling line localization is new to our knowledge. It is demonstrated in Section 3.2 that the methodology which is designed for this task is able to correctly identify the ruling structure with sufficient accuracy. Moreover, the feature design which advances existing texture features for the task of ruling classification, represents supporting material correctly even if a page is copied multiple times which results in broken ruling lines. Having detected a ruled page, ruling lines are located with sub-pixel accuracy which is used as alignment feature for document reassembling. The line localization is based on Projection Profile (PP) and incorporates a-priori knowledge such as a global ruling line spacing and parallelism of ruling lines up to a certain degree. By these means occluded (e.g. by text) ruling lines are reestablished even if they are not visible to the human observer anymore. In addition to the application of ruling analysis for document grouping and document snippet alignment, an evaluation is performed which demonstrates that the ruling analysis is capable of removing ruling lines from binary images with sufficient accuracy. The ruling line removal is evaluated on a synthetic database which is 6 and 12 times larger than that of other state-of-the-art methods [AKD09; LK10]. Although, the methods cannot be compared directly because of the lack of a standardized database, the results gained by the method proposed, are promising. The synthetic database which is created for this task is made available to the public such that other state-of-the-art methods can be directly compared to the ruling line removal proposed.

Text Classification: Before classifying text into handwritten, printed, and non-text elements an element-wise localization is performed. The localization is based on a binary image which labels elements into foreground and background. In order to keep the representation of words compact while accounting for typical distortions such as local skew or elongated ascenders/descenders so called Profile Boxes are introduced. In contrast to well-known representations such as the Bounding Box (BB) or Minimum Area Rectangles, Profile Boxes do not enclose the binary representation of words. This property allows for an accurate localization that is not biased if handwriting is present. Furthermore, the layout analysis in the presence of handwriting is improved, since text line merges are reduced by the compact representation.

Problems entailed by binarizing multimodal and degraded documents are handled during text classification. First, the feature extraction is carried out on the gradient image which is computed from the gray-scale image. Second, an additional non-text or *noise* class is introduced which rejects false positives of the binarization. The idea of introducing a noise classification which bears binarization errors is not new [ZLD04]. However, an efficient learning strategy of noise elements is presented.

In contrast to state-of-the-art text classification methodologies (e.g. [Pal+07; Kan+07; KKH08; KH12; Han+13]) which classify text using typical document analysis methodologies, the system proposed uses modern computer vision approaches for text classification. Therefore, GSFs are newly introduced that are successfully applied to various document analysis scenarios including medieval manuscripts (*Saint Gall* database), modern printed documents (PRImA database), handwritten documents (IAM Database [MB99] (IAM-DB) and Computer Vision Lab Database [Kle+13] (CVL-DB)) and the Stasi snippets. These features which are inspired by Shape Context (SC) [BMP02] and local descriptors such as Scale Invariant Feature Transform (SIFT) [Low04] are capable of being applied to various computer vision scenarios because they keep a low dimensional representation while being robust with respect to degradations and deformations of objects.

Layout Analysis: A new bottom-up layout analysis is presented which groups text elements based on their Profile Boxes. The text line clustering is based on global energy minimization with respect to three rules which are detailed in Section 5.1. These rules allow for a flexible text line grouping which is robust with respect to local skew, local deviations of the word's heights or slant and noise. Furthermore, a grouping is possible even if documents lack a global layout structure which is likely for torn documents. In contrast to state-of-the-art page segmentation algorithms such as the participants of the Handwriting Page Segmentation Competitions [GAS07; GSL09; GSL10; Ant+13] who assign per-pixel text line labels, the methodology proposed aims at finding an optimal rectangle which accounts for a text line's local skew. Again, the extraction does not detect an enclosing rectangle but rather finds a rectangle that fits the words' *x-height*. By these means, the text line localization is used on the one hand to group possible neighbor pieces and on the other hand to accurately align matching snippet candidates. Even though, the application is different from those of typical page segmentation algorithms,

it is demonstrated in Section 5.3 that it can compete with state-of-the-art methods.

All contributions are summarized in the list below:

- **Supporting material classification** is a new approach which allows for improving ruling line localization and grouping document pages based on their ruling structure.
- **Ruling line removal database** is a database which is synthetically generated and made available publicly so that state-of-the-art methods can be compared to the line removal proposed.
- **Profile Box** is a word representation which represents words by their *x-height* and local skew rather than enclosing the whole CC.
- **Gradient Shape Features** are newly introduced gradient features which render text classification possible in heterogeneous document analysis scenarios.
- **Training of non-text elements** is improved by treating noise elements differently during training of the machine learning algorithm.
- **Text line clustering** is carried out on Profile Boxes with newly introduced rules which allow for text line clustering even if sparse text data is present.
- **Visual content extraction** of document snippets is the combination of all document analysis tasks presented. It allows for a rich representation of document snippets which is used for document clustering and document reassembling.

1.3 Definition of Terms

To clarify terms which may be used differently from their common intent, a definition of terms is given in this section. These terms include words commonly used such as *document snippets* but also words newly introduced that specify the result of an algorithm or the algorithm itself.

Document snippet is referred as document parts of digitized documents which are resulting from manually tearing documents. They are assumed to have a mean area of 42.4 cm^2 with a standard deviation of $\sigma \pm 37.1$ cm^2. The content ranges from no text at all over a few single words or text lines up to fully inscribed DIN A4 pages. Figure 1.4 shows three samples of typical document snippets. Because of the semi-automated scanning process, snippets have mutual dominant orientations which is determined in a preprocessing step. Furthermore, the manual tearing process leaves noise at the snippet's border. It is shown that the content of snippets varies from mixed environments a) over sparsely inscribed document snippets b) to void snippets c).

Ruling Analysis refers to the automated ruling classification and localization. *Ruling* includes solely those lines in a page that are pre-printed for guidance of handwriting. These lines typically have low contrast, are parallel to each other and have a certain

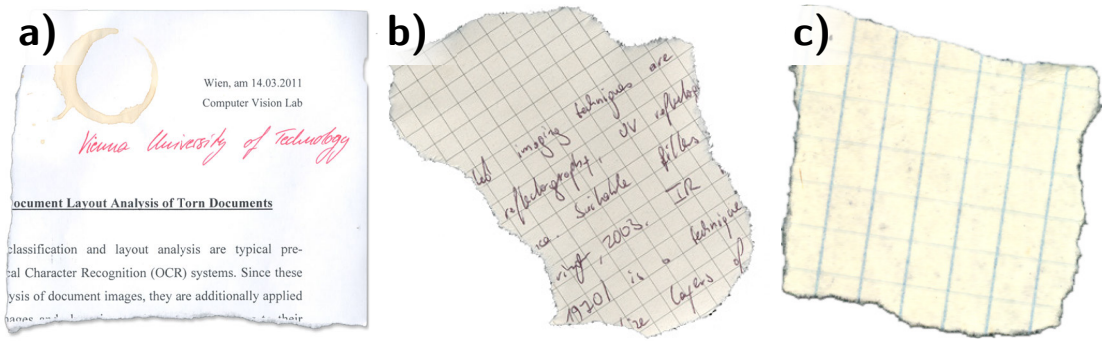


Figure 1.4: Three document snippet samples. Note that a) and b) are sample snippets which reflect typical challenges since the original Stasi reports must not be published.

spacing. Hence, lines which emphasize text or compose tables and forms are not regarded as *ruling*. This strict definition results in classification errors since table lines have the same properties such as parallelism and fixed spacing. However, ruling is the only property that is assumed to be present in all snippets of a document page which allows for grouping document snippets based on this criterion – in contrast tables do not necessarily span a whole document page. The ruling analysis is split into three processing stages. First, the supporting material is classified into *void*, *lined*, or *checked*. Then, ruling lines are roughly localized by means of a PP. Finally, an accurate line localization is performed which accounts for slight local deviations.

Profile Box is a word representation which is newly introduced. Profile Boxes approximate a word's *x-height* rather than enclosing the word's CC. Figure 1.5 illustrates three different approaches for approximating a word with a rectangle. The illustration in a) is a Bounding Box (BB) which is frequently used in the literature for word approximation. The second illustration b) shows a minimum area rectangle that finds a CC's enclosing rectangle with minimal area. It is illustrated in this sample that the minimum area rectangle does not necessarily account for a word's local skew. The last example c) shows a Profile Box which detects the word's local skew while keeping a compact representation. Profile Boxes are computed by robustly fitting lines into the component's upper and lower profiles.

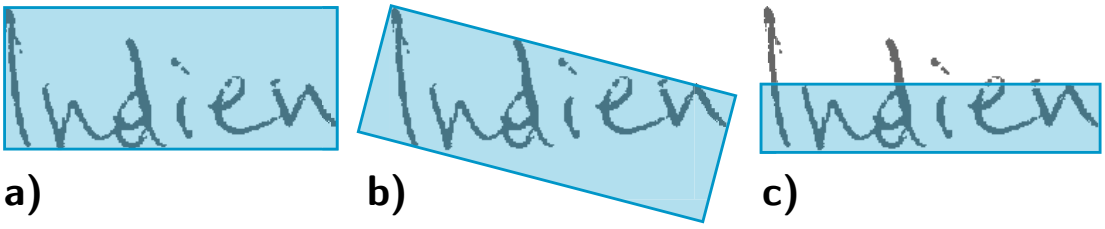


Figure 1.5: A bounding box which covers the whole CC a), the corresponding minimum area rectangle b) and the Profile Box proposed c).

Gradient Shape Feature is the feature which is newly introduced for text classification. These features account for edges by accumulating the gradient magnitudes to a log-polar grid. In contrast to other state-of-the-art local features, GSFs are robust with respect to scale changes by normalizing their grid size with respect to the height of Profile Boxes. Furthermore, rotation invariance is achieved by determining a word's skew. The feature extraction is additionally robust with respect to binarization errors since the gradients are extracted rather than a word's contour.

1.4 Results

All three methodologies are evaluated on the Stasi database which is discussed in more detail in Section 4.5.4. The methodologies are evaluated on additional scientific databases which allow for drawing conclusions between the system proposed and state-of-the-art document analysis systems. Figure 1.6 shows six samples of the databases used for empirically evaluating the performance gained by the methods presented. A sample from the ruling database synthetically generated can be seen in a). The text classification is evaluated on the PRImA (b), the Stasi (c) and the IAM-DB (d) databases. The last two samples are from the ICDAR 2013 Handwriting Segmentation Contest (e) and the *Saint Gall* database (f) which are used for evaluating the layout analysis.

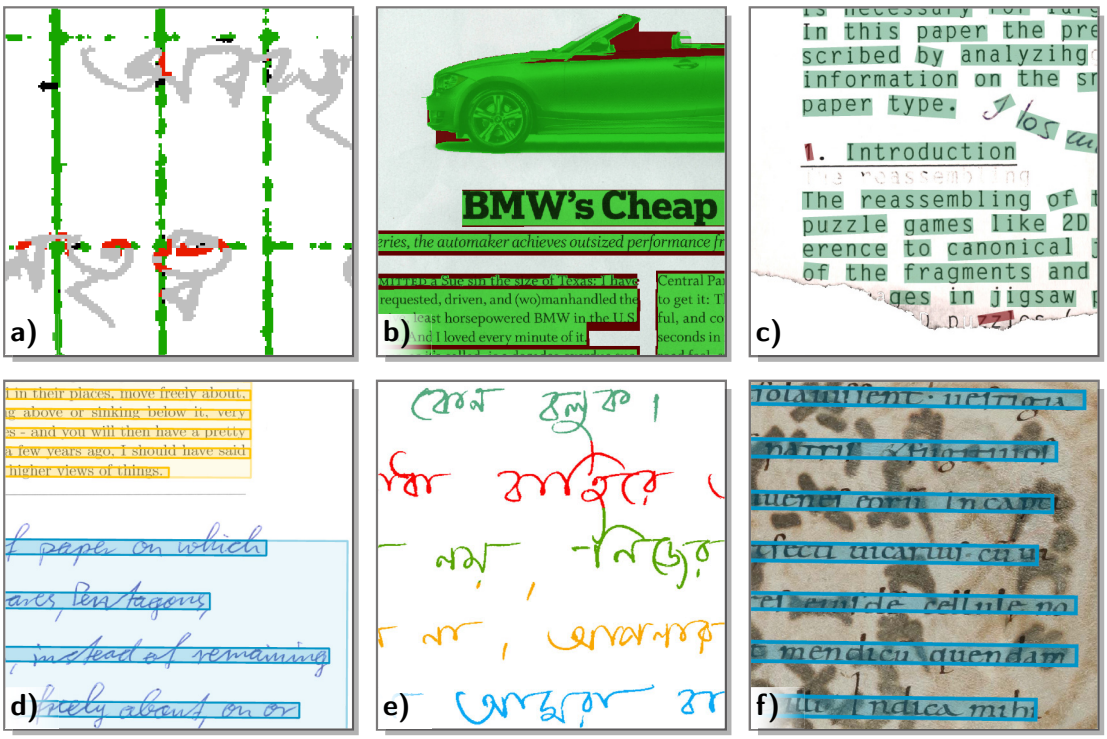


Figure 1.6: Samples from databases used for evaluation.

1.4.1 Ruling Analysis Evaluation

The ruling classification is evaluated on two sets of torn Stasi documents. One of which contains all document snippets annotated. The second set is a clean set which does not contain any ambiguities such as lines from tables or forms. Both sets allow for drawing conclusions of the ruling analysis performance with and without the presence of obscured data. On the clean set an F-score of 0.987 is achieved, while the F-score on the degraded set is 0.919. On this set the precision of *lined* pages is especially reduced since they are likely to be misclassified as void paper. This effect can be attributed to ruled pages which are copied at high contrast and therefore contain high contrast ruling lines. The classifier confuses these lines with table ruling or form ruling.

Parameters that alter the system's behavior are empirically evaluated on the Stasi database. The parameter values which maximize the F-score are finally used for classification. In order to compare the system's performance with state-of-the-art ruling analysis methods, the ruling removal quality is examined. Therefore a synthetic database is generated. The database of the ICDAR 2013 Handwriting Segmentation Contest which consists of 150 handwritten documents serves as basis. Four degraded ruling templates (where four is a trade-off between variety and database size) are added to the document pages resulting in 600 documents. Since *noise* is added synthetically, the ground truth is known implicitly. Then, the proposed ruling analysis is used for automatically removing the ruling lines which is empirically evaluated on all pages. The system achieves an F-score of 0.93 which is the best ruling line removal performance found in the literature.

1.4.2 Text Classification

The text classification performance is evaluated on five databases. The first database (PRImA) was the basis for the ICDAR 2009 Page Segmentation Competition [Ant+09a]. An evaluation on this database demonstrates the system's performance if visual content of modern printed documents is analyzed. For this evaluation, the text classification is trained on images and diagrams. The system achieves an F-Score of 0.945 which is about 1% better than the best participating method of the competition. An evaluation on more recent Page Segmentation Competitions (e.g. [Ant+13]) is not performed since these databases contain orthogonal text blocks which cannot be handled by the system proposed.

The second evaluation on the IAM-DB and CVL-DB demonstrates the performance gained on predominantly handwritten documents. This evaluation further compares the proposed system with four state-of-the-art text classification approaches. It is shown that the system can compete with these approaches even though the evaluation is carried out on all 1534 pages. All other methods compared are evaluated on maximal 103 pages. An overall F-Score of 0.998 is achieved in this evaluation scenario.

In addition to the evaluations which allow for comparing the proposed system with other systems, the performance is tested on real world data from the Stasi database. Since this database consists of real world Stasi records with sensible content, the data-

base and its images must not be published. That is why, illustrations in this thesis use replicated document snippets that have similar attributes as the original records. Intuitively, the F-score gained on real world data is lower than that achieved on the databases mentioned previously, because these documents are torn, have highly varying supporting material and content. On this database an F-score of 0.899 is achieved. Especially bleed-through text lower the precision of *noise* elements. In addition to the system evaluation, tunable parameters (see Section 4.5.5) are empirically evaluated on the Stasi database.

1.4.3 Layout Analysis

The layout analysis is evaluated on the databases of the three most recent Handwriting Segmentation Contests [GSL09; GSL10; Ant+13] in order to allow for comparisons of the proposed layout analysis with other state-of-the-art methods. On the most recent contest a pixel based Performance Metric (FM) of 0.966 is achieved which is by 2% worse than the best performing method that participated in the ICDAR 2013 Handwriting Segmentation Contest. A more detailed evaluation which is discussed in Section 5.3.3 shows that the inferior performance can be attributed to a low precision of handwritten Bangla text. In order to improve the system for this objective, rules for correctly grouping accents would need to be implemented. Furthermore, the labeling applied for pixel accurate text line segmentation is not optimal.

The second layout analysis evaluation is performed on the *Saint Gall* database [Fis+10] which are scanned document images of a medieval manuscript. An FM of 0.99 is achieved on this database which outperforms other state-of-the-art methods.

Thesis Structure

Related work from the document analysis community is presented hereafter in Chapter 2. First methods for ruling line removal are depicted in Section 2.1. Then, features for text classification are discussed in Section 2.2. Since machine learning is needed for text classification a brief overview of machine learning methods commonly used in document analysis is given in Section 2.3. Even though, machine learning is needed for accurate text and ruling prediction, it is not a part of investigation of this thesis. A comprehensive overview of text classification systems is presented in the proximate Section 2.4. Finally, related work in the field of page and text line segmentation are discussed in Sections 2.5 and 2.6 respectively.

The ruling analysis is detailed in Chapter 3. First the methodology including patch extraction (Section 3.1.1), feature design (Section 3.1.2), classification (Section 3.1.3) and the ruling line estimation (Section 3.1.4) is discussed. In Section 3.2 a thorough empirical evaluation is performed. The evaluation compares the ruling line removal performance to other methods proposed in the literature and analyzes errors and parameters on real world data.

Text classification is described in Chapter 4. This chapter is further disposed into a short discussion of pre-processing steps needed for robust text classification (Section 4.1).

Then, the text localization which details the idea of Profile Boxes is given in Section 4.2. Having localized text components such as words or characters, GSFs – depicted in Section 4.3.3 – are extracted. The training and classification is given in Section 4.4. Again, results of evaluation on public databases and on the Stasi database allow for drawing conclusions of assets and drawbacks of the methodology proposed. The results are presented in the last section (4.5) of this chapter.

The last chapter dealing with the system proposed addresses layout analysis. There, the text line clustering is presented in Section 5.1. Subsequent to text line clustering the localization is discussed in Section 5.2. An evaluation performed on the Page Segmentation Contests (Section 5.3) compares the layout analysis proposed with state-of-the-art page segmentation methods. The thesis is concluded in Chapter 6.

CHAPTER 2

Related Work

Document Analysis and Understanding is an intense research topic regarding the amount of publications (see Figure 2.1) in the three major international conferences (ICDAR, ICFHR, DAS) and the scientific journal IJDAR dedicated to document analysis. The objective is to automatically extract textual information from document or natural scene images and thus introduce a higher level description that is further automated or processed. At first glance OCR and handwriting recognition are the ultimate goals of document analysis. However, there are other incentives in document analysis which are not linked to the transcription of documents. Beside the topics layout analysis [KSI98], text line segmentation [KC12], and text classification [ZLD04] which are related to this thesis, there are applications such as writer identification [FS13], signature verification [Zhu+09], graphics recognition/understanding [SL13], word spotting [Fri+12; Fis+12], document classification [GPV13], and document clustering [GPV13].

In order to quantize the scientific effort in this field, a statistics of the most relevant international document analysis conferences is given in Figure 2.1 and Table 2.1[1]. It can be seen that ICDAR is the largest conference in this field and that the amount of scientific papers is slightly increasing over the years. In contrast to general computer vision conferences (e.g. the CVPR which was first organized in 1983), document analysis conferences started out in the beginning of the 90s. In Table 2.1 the $h5$-Index and the $h5$-Median[2] of the conferences are given which represent the scientific impact. The CVPR which has the highest ranking in computer vision is additionally listed to allow for a direct comparison between computer vision in general and the subtopic document analysis. The $h5$-Index indicates the number of a conference's publications h in the past 5 years (2008-2012) that were cited h times. The $h5$-Median is the citation count median of these h papers. Both values reflect on the one hand the scientific impact of a conference as they indicate a citation count and on the other hand the size of a conference (Note a conference which accepts 50 papers in total can maximally have an

[1] The statistic is based on summaries of the proceedings, `www.dblp.org` and `iapr-tc11.org`

[2] scholar.google.com (2013, November 21)

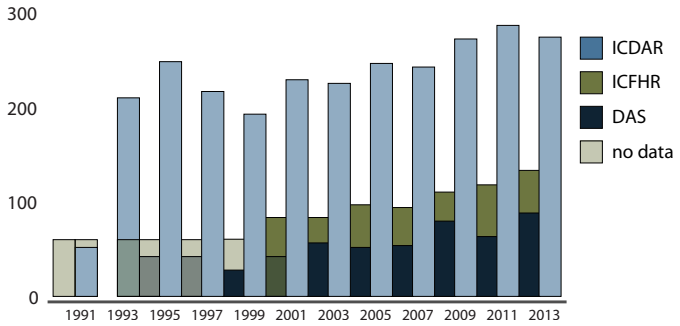


Figure 2.1: Number of publications of the three international document analysis conferences.

$h5$-Index of 50).

Due to its modality, document analysis has a few pre-processing steps that are not used for general object recognition pipelines. If text needs to be extracted from documents, a binarization is typically applied since binary entities simplify processing (e.g. CCs) [KDS11; PGN13]. Since binarization gets challenging if noisy data is present (e.g. historical documents), binarization free approaches are presented [Yos+09; ZL12; Gar+13] or the recognition step incorporates models that can distinguish between binarized noise and text [ZLD04]. Fast grouping techniques such as PPs or anisotropic Gaussians are sensitive to skew. In order to extract document information without the need of rotation invariance, a preceding skew estimation is performed that allows subsequent processing steps to be sensitive to skew changes [DKS12; Pap+13].

These pre-processing steps are not further discussed in this section. Instead, a focus of related work in the field of *Ruling Analysis, Text Classification, Layout Analysis*, and *Text Line Segmentation*, which are the topics of this thesis, is gained. State-of-the-art text classification systems have all in common that features are first extracted which are then classified into either *printed* or *handwritten*. Depending on the data, other classes such as *graphics* or *line drawing* are used in addition. Because of this system architecture, features which are used in modern text classification applications are introduced first. Then, a short introduction of machine learning algorithms is given in Section 2.3. An exhaustive review of text classification systems with a comparison of

	1st Issue	Papers	h5-Index	h5-Median
CVPR	1983	328	106	174
ICDAR	1991	234	24	29
IJDAR	1998	22	17	23
DAS	1994	57	15	20
ICFHR	1990	87	9	10

Table 2.1: Comparison of Document Analysis conferences.

their scientific impact can be found in Section 2.4. Extending the idea of text classification, Layout Analysis systems are presented in Section 2.5. Finally, related work in text line segmentation is detailed in Section 2.6.

2.1 Ruling Analysis

Ruling analysis is proposed for applications such as music score analysis [RC13], form analysis [Zhe+01; ZLD05], or handwriting processing [LK10; CL14]. The former two applications analyze ruling for further analysis or segmentation tasks while in the latter ruling is assumed to be noise, needed to be removed in a pre-processing step [AKD09; LK10]. If ruling lines in handwritten documents are considered, a few properties can be derived as presented by D. Lopresti and E. Kavallieratou [LK10]. Ruling lines have a uniform contrast, thickness, orientation, and spacing. They are – on purpose – brighter than the writing which overlaps with ruling. These a-priori assumptions are incorporated in ruling algorithms to increase their performance.

Y. Zheng et al. [Zhe+01] extracted ruling lines in forms using Directional Single-Connected Chains (DSCC). They analyze white runs in binary images in order to detect lines. *Abnormal* run-lengths and crossings are removed so that lines which are merged with other foreground elements (e.g. text) get isolated. A dynamic length threshold, which is estimated based on the character size, rejects short lines such that lines of characters are not removed. This methodology is fast and able to extract clean lines present in document images without degrading foreground elements. However, broken ruling lines that occur due to the properties previously mentioned cannot be extracted using the DSCC. In order to overcome this drawback, Y. Zheng et al. [ZLD05] extended the algorithm. In this approach they again detect lines in the binary image using DSCCs. Then, a Hidden Markov Model (HMM) is trained on the document's PP for accurate line positioning. Finally, lines are represented by polylines which account for non-rigid distortions present due to bad storage or digitization conditions.

W. Abd-Almageed et al. [AKD09] present a ruling line removal for handwritten documents. For each pixel in the binary image features are computed within a local neighborhood based on central and statistical moments (standard deviation and kurtosis). The pixel are then classified into *line* or *non-line* pixel using a linear subspace. Though this algorithm has a potential for general line detection scenarios (no assumption about line parallelism or uniform spacing is incorporated), it tends to falsely classify stroke endings of characters, ruling lines close to text and noisy lines.

D. Lopresti and E. Kavallieratou [LK10] present a ruling line removal based on a scanning approach. Lines are initialized in the left and right area of a page and the mean line slope and thickness are derived. Then, each column is scanned for pixel groups that have a similar thickness and are potentially on a line that was detected during initialization. The line slope and position is iteratively updated to account for quantization errors. Recently, J. Chen and D. Lopresti [CL14] present a ruling line detection for handwritten documents. A rough estimation of potential ruling lines is made by a simplified Hough transform. Then, broken line segments are clustered using

Sequential Clustering. A multi-line linear regression finds a global optimum in the sense of Least Square Error (LSE) of ruling lines.

2.2 Features for Text Classification

In this section features that are used to discriminate handwritten from printed text are presented. All of them except for gradient features and SIFT are extracted from binary images. This is specific for document analysis as general object recognition methodologies tend to use features extracted from gray level images. The design choice is related to the fact that documents generally have two classes (*text, non-text*) which renders binarization challenging but possible (see DIBCO [PGN13; PGN11]).

The section is structured as follows. First handcrafted features which are specialized to properties of specific fonts are presented in Section 2.2.1. Layout features which are designed to capture relations between elements are presented in Section 2.2.2. Then statistical features including PP features (Section2.2.3), moment invariants (Section 2.2.4) and co-occurrence features (Section 2.2.5) are presented. Finally histogram features are described.

2.2.1 Handcrafted Features

In this section, handcrafted features are summarized. The term *handcrafted* in this context refers to the design process of features which are in this case specialized to a specific set of problems (e.g. features for sans-serif fonts).

Khunke et al. [KSK95] propose features that extract a character's line straightness and its symmetry. These features consist of four dimensions. Two of which represent the straightness of vertically and horizontally oriented lines. The other two dimensions are so-called symmetry features. First they detect lines in a local image patch by means of a Hough Transform which is quantized at 16 directions. Then, a line is fit to the pixels by means of regression. The straightness is finally computed by taking the variance σ^2 of the pixels with respect to the fitted line.

The symmetry features are computed for the contour detected and inner loops with respect to the center of gravity. The latter is set to zero if a character does not have any inner loops. The symmetry is defined to be one when a foreground pixel, which is reflected with respect to the center of gravity, is again matched with a foreground pixel. After accumulating and normalizing all possible symmetries, a value close to one indicates "perfect symmetry" while characters with no symmetry at all have a low value.

2.2.2 Layout Features

In contrast to all other features presented in this chapter, layout features incorporate the relation between two or more elements such as CCs or words. All other approaches extract features on character, word, or line level and use the relation between words in a post-processing step (e.g. [ZLD04]).

The Character Block Layout Variance (CBLV) features [FWT98] assume differences in character or word layout that arise from differences in generating machine printed and handwritten characters. First BBs are computed either for characters or words – if the characters are connected. The layout of characters or words is assumed to be regular if the BBs lie on a straight line with respect to a certain threshold. Then, text blocks are classified by applying a second threshold on the normalized CBLV features:

$$CBLV = \frac{\sum_{i=1}^{N-1} C(i)}{N} \tag{2.1}$$

where N is the number of BBs in a text block and $C(i)$ is defined to be zero for regular layouts and one otherwise. Fan et al. define regular layouts if the distance between the central bottom points of two subsequent BBs is shorter than the tenth of the median BB height.

Another feature that is designed to discriminate between handwritten and printed Bangla or Devanagari scripts is the so-called Character Lowermost Point Standard Deviation (CLPSD) [PC99; PC01]. It is similar to the CBLV, but the lowest points of a component are regarded rather than the bottom of bounding boxes. The standard deviation of these points is computed with respect to the baseline and "lower line" (this line fits the descenders) of characters. According to Pal and Chaudhuri, the feature's value is close to zero for machine printed text.

2.2.3 Features based on Projection Profiles

A PP is computed by accumulating the pixel values along a specific dimension. For document images, they have the advantage, that the text's x-height, ascenders, and descenders can be easily retrieved. If global projection profiles are used, their performance is crucial with respect to slight changes in the document's skew.

Guo and Ma propose PP features [GM01]. They compute the vertical PP of each character's BB. The PP is a histogram along a specific dimension. In other words, a vertical PP is computed by accumulating all black runs (foreground) in y direction. In order to be robust with respect to scale changes of characters, Guo and Ma propose to normalize the PP such that its sum is equal to 1. Then, they quantize each PP bin into 10 different levels.

Zheng et al. [ZLD01; ZLD04] extend PP features which they call run-length histogram features. First, they compute a PP along the horizontal, vertical axes, major, and minor diagonal which allows them to capture slanted text more accurately. Then they normalize the histogram in order to get scale invariance. After this step, the number of histogram bins corresponds to the number of pixels along the axis sampled. In order to implicitly normalize characters, they propose to quantize the PP histogram into five bins. The quantization is useful to correct for varying character widths in handwriting, but also for printed text when *proportional* typefaces are used rather than *monospaced*. According to Zheng et al., the features are more robust if overlapping Gaussian kernels – rather than rectangular windows – are used for quantization.

2.2.4 Moment Invariants

Moment invariants are statistical features that are extracted by means of higher order moments. The two dimensional moments of a density distribution $\rho(x, y)$ can be defined as Riemann integrals [Hu62]:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \rho(x, y) dx dy \tag{2.2}$$

for a density distribution $\rho(x, y)$ and order p, q. In image processing central moments are used in order to guarantee translation invariance. Hence, moments are computed relative to a distribution's centroid:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y) \tag{2.3}$$

with $I(x, y)$ being the image (or CC) and $(\bar{x}, \bar{y})$ its centroid. Hu [Hu62] defines seven nonlinear functions on central moments that are translation, scale, and rotation invariant. These moments are frequently used in pattern recognition tasks [Flu02]. In 1990, Khotanzad et al. [KH90] propose the use of Zernike moments which have an orthogonal basis in contrast to the Hu invariants. Hence, Zernike moments do not have any redundant information, which is an important property for image reconstruction. However, Jan Flusser [Flu00; Flu02; Flu05] shows that Zernike moments are not independent which is fundamental for features. Flusser introduces complex moments and shows how to derive invariant sets that are independent. In 1993 Flusser and Suk [FS93] introduce four affine moment invariants. In contrast to Hu's invariants which are invariant with respect to translation, rotation, and scale, the affine invariants do not change under any affine transformation. On the basis of Flusser's findings, Rahtu et al. [Rah+06] propose a general object recognition approach using affine invariant moments. Despite Flusser's findings the Hu invariants are still used for text classification [Kan+07; Han+13].

2.2.5 Co-Occurrence Features

The co-occurrence matrix is introduced by Haralick et al. [HSD73] for texture classification, which they called *Gray-Tone Spatial-Dependence Matrices*. A co-occurrence matrix counts equal pixel values at a certain distance. Equation 2.4 illustrates the vertical co-occurrence count for binarized images.

$$C_h(d) = \sum_x \sum_y I(x, y)\ I(x, y + d) \quad d = 1, ..., N \tag{2.4}$$

where $I(x, y) = 0$ is defined as background. Zheng et al. [ZLD04] propose to use four distances ($N = 1, 2, 4, 8$) and four orientations ($\pi/4$). Since background pixels in document images are more common ($> 80\%$) than foreground pixels. That is why, they propose to sample solely foreground co-occurrences i.e. black-black pairs. In order to achieve scale invariance, co-occurrence features are normalized so that the sum of each distance d is one.

Zhang and Lu [ZL12] propose Edge Co-occurrence Matrix (Ecm) which they extract in sampled block windows. They claim, that the stroke width is not needed to tell machine printed text from handwritten text apart. Hence, they compute the co-occurrence matrix for block windows which were previously filtered with a Sobel edge filter. They account for orientation changes by detecting the main orientation. Therefore, the Ecm is computed for repeatedly rotated block windows. The orientation which maximizes the distribution is assumed to be the main orientation.

Aya Soffer [Sof97] proposes $N \times M$-grams which are an extension of N-grams. First, the patterns of an image are extracted using a sliding window with a size of $N \times M$. Then, sliding windows with the same pattern are accumulated resulting in a feature vector size of $2^{3\times3} = 512$ for a 3×3-gram. This feature vector is finally normalized to one. Doermann and Liang [DL01] extend this approach. They solely extract 2×2-grams from a text block (e.g. character). In contrast to Soffer they extract the 2×2-grams at multiple scales (distance $d = 1, 2, 4, 8$) without increasing the pattern size (see Figure 2.2). Hence solely the four corner pixels are computed which keeps the resulting feature vector comparatively small ($2^{2\times2} = 16$) but still local patterns are extracted at different scales. $N \times M$-grams that solely contain background pixels are rejected in their approach so that background is not over emphasized. In addition, they rather normalize the resulting occurrence vector by the density of the currently observed text block than normalizing it to one. In doing so, the resulting feature vector reflects the occurrence of $N \times M$ grams with respect to the total number of black pixels.

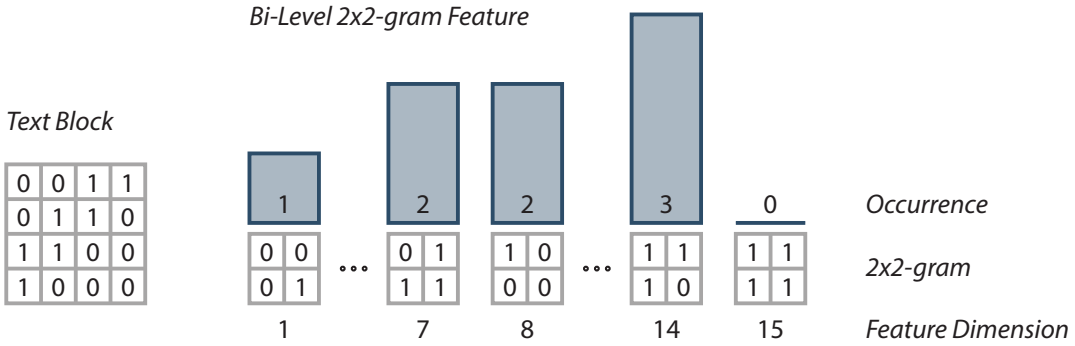


Figure 2.2: Sample text block with its corresponding 2×2-gram feature ($d = 1$) as proposed by Doermann and Liang [DL01].

2.2.6 Chain Code Histogram Feature

Chain Code Histogram (Cch) features are introduced by Pal et al. [Pal+07] and successfully applied to text classification by Chanda et al. [CFP10]. The feature design is closely related to those of Sift with the difference that Chain Codes are used rather than gradients and the scale normalization for Cch is implicitly achieved by segmen-

tation. In order to extract CCHs, a CC's bounding box is divided into 7×7 blocks. For each block, the chain code frequency is recorded i.e. the four different orientations $(-\pi/4, 0, \pi/4, \pi/2)$ are accumulated and normalized by the maximal bin. Then, the spatial grid is reduced from 7×7 to a 4×4 grid using a Gaussian weighting which results in a 64 dimensional feature vector. Figure 2.3 shows the computation of CCH features. First the Chain Code b) is extracted for all contour pixels. Note that solely orientations between $(-pi/2\ pi/2]$ are accumulated (e.g. *up* (2) is equal to *down* (6)).

Figure 2.3 illustrates the 7×7 block grid which is fit into a CC's bounding box. For each block, the chain code distribution is computed in d). Finally, the grid is down sampled to a 4×4 grid using a Gaussian which is shown in e). The resulting feature vector has $4 \cdot 4 \cdot 4 = 64$ dimensions.

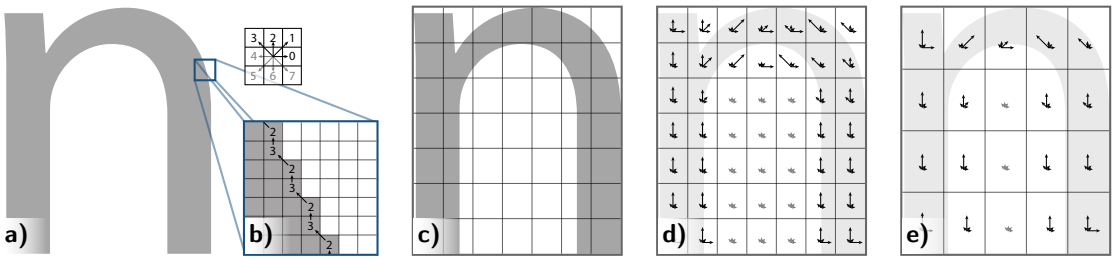


Figure 2.3: Computation of CCH features. The Chain Code is computed for contour pixels in b). c) shows the 7×7 grid of the BB. Chain Code density vectors are illustrated in d) and e).

2.2.7 Transform Features

This section summarizes transform based features. The term *transform* rather refers to the transformation between two mathematically defined spaced (e.g. DFT) than a simple affine transformation of the image which is generally needed in the feature design. The transforms presented have the positive property that they emphasize certain structures (e.g. stroke width, stroke orientation) that can be used for differentiating between handwritten and machine printed scripts.

Koyama et al. [KKH08] propose the use of power spectrum features. They claim that these features in combination with an MLP simulate the human vision. First, they transform sliding windows having a fixed size $W \times W$ by means of the FFT (see Figure 2.4). Then, the Fourier spectrum is shifted such that low frequencies are located at the sliding window's center. After this transform, strokes generate local maxima along their main orientation. That is why, Koyama et al. propose to sample the power spectrum with respect to 16 predefined angles:

$$E_i = \frac{\sum_{|v - tan(\theta_i)u| \leq 1} S(u,v)}{\sum_u \sum_v S(u,v)} \tag{2.5}$$

with $S(u, v)$ being the power spectrum of the sliding window and θ_i the angle of the i-th axis. The normalization term in Equation 2.5 guarantees that the features are scale invariant and robust with respect to luminance changes. Features generated that way are not rotationally invariant. Koyana et al. do not mention how the choice of the sliding window's size W affects the classification performance if characters with different sizes are present.

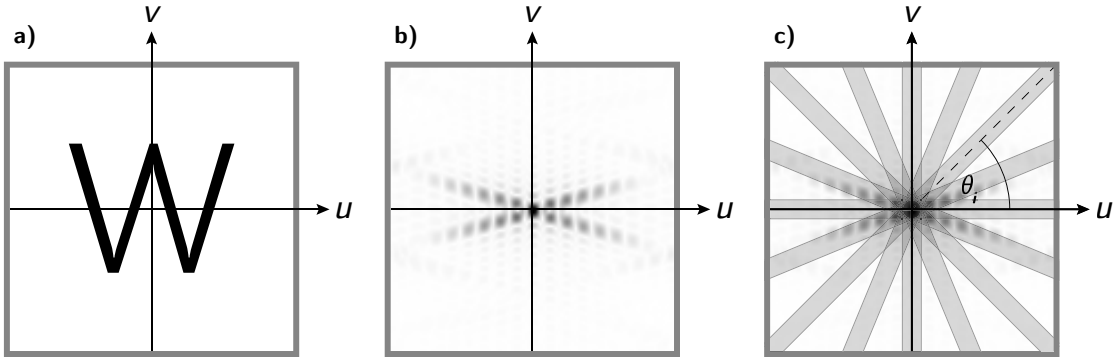


Figure 2.4: Sliding window of a character a), FFT of the character b) and the 16 regions extracted c) (see Equation 2.5).

Radon Features Zemouri and Chibani [ZC11] introduce Radon transform features for text classification. The Radon transform, named after the Austrian mathematician Johann Radon (1887 – 1956), is a pendant to the Hough transform with the difference that continuous functions (lines) are transformed rather than binning the discrete gradient vectors. Hence, it transforms an image space to a two dimensional space where the abscissa represents all line angles possible (0 2π] and the ordinate represents ρ which is the distance between a line and the origin. In order to compute Radon features for text classification, words are segmented in a document image. Each word is then transformed by means of the Radon transform. Then, a PP of the squared values is computed along the abscissa which represents different angles θ of lines extracted in the word image. Strong peaks (values) in the resulting 360 dimensional feature vector indicate a higher frequency of lines having the corresponding angle. Hence, handwritten text tends to have broader peaks than machine printed since more different stroke angles are present. The features are normalized using a non-linear transform which is unfortunately not detailed by Zemouri and Chibani. Figure 2.5 shows the Radon features of a printed and handwritten word. The word is transformed using the Randon Transform in b). Then, the squared PP is computed for each word c).

This approach is further extended by Konno and Hirose [KH12]. They apply the Hough transform to a sliding window due to computational reasons. Having accumulated the histogram, they compute local statistics for an interval of $\pm 7°$ at the positions $-\pi/2$, $-\pi/4$, 0, $\pi/4$, $\pi/2$. In doing so, they can further reduce the feature dimension.

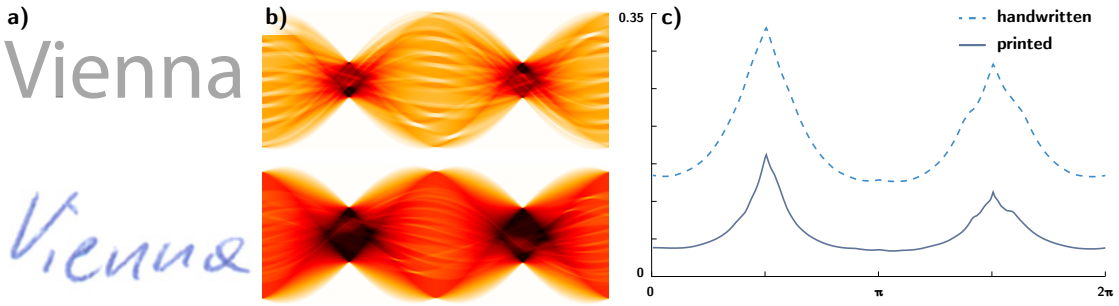


Figure 2.5: Radon transform of the printed and handwritten word respectively b). Randon features c) [ZC11].

2.2.8 Gradient Features

Gradient Features are extracted from gray-scale images without a prior binarization. Both features presented in this section extract information by means of gradients. Gradient vectors are frequently used for information extraction since they have a low magnitude for homogeneous regions and a high magnitude at edges [HS88]. Though SIFT is computed by means of gradient vectors too, it is detailed in Section 2.2.9 which addresses local descriptors.

The Imade feature [ITW93] combines a luminance histogram with a gradient vector histogram. Both are sampled in a randomly chosen 32×32 px window. The gradient vector's angle θ of each pixel is accumulated to a 24 bin histogram. Hence, each bin accumulates the gradient angles of 15°. The gradient vectors are neither weighted by the gradient magnitude nor by their respective location within the current window.

The luminance histogram is defined as a 32 bin normalized histogram of the observed window. In order to design features robust with respect to images with poor dynamic range or illumination changes, they normalize the luminance histograms such that 5% outliers in either direction (dark, bright) are cropped. Additionally, the histograms are normalized with respect to their maximal bin. Figure 2.6 illustrates the features for the four classes that are defined in [ITW93]. Note that printed text has high bins for the orientation bins that are parallel or perpendicular to the document's main orientation, while photographs have a uniform luminance distribution and no significant gradient orientation bins.

Gabor Filters Gabor Filter features presented in [ZLD02; ZLD04] are a set of features that capture a document's local structure. The Gabor filter is an edge filter that is combining the Gaussian filter with a sigmoid function:

$$g(x,y) = \exp\left(-\pi\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)\right) \cdot \cos\{2\pi(u_0 x + v_0 y)\} \tag{2.6}$$

where $x' = -x\sin\theta + y\cos\theta$, $y' = -x\cos\theta - y\sin\theta$, $u_0 = f\cos\theta$ and $v_0 = f\sin\theta$. f is the frequency and θ is the orientation. The advantage of the 2D Gabor function

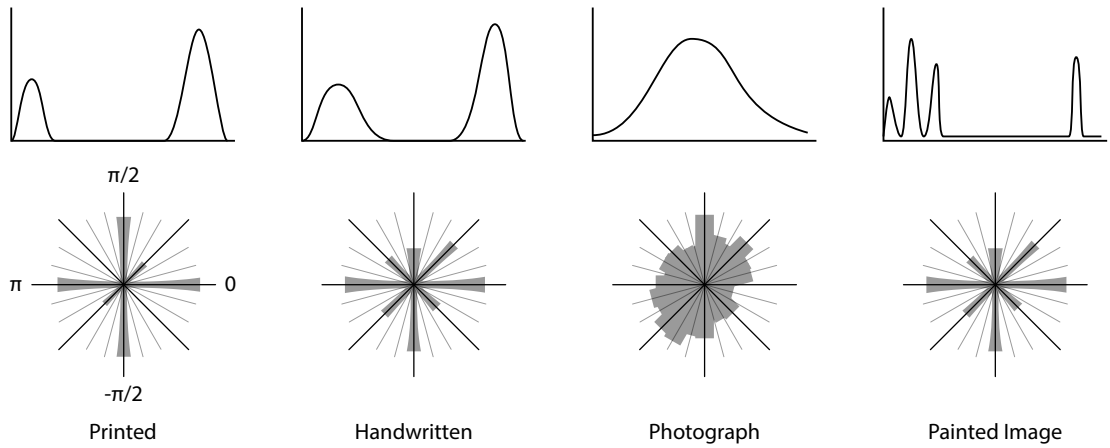


Figure 2.6: Schematic illustration of the Imade features. The upper row shows the luminance features for all four classes respectively while the lower line illustrates the gradient features.

compared to Gaussian derivative kernels is its parameterization. Figure 2.7 shows an even Gabor filter bank. In this case θ is chosen to be $\theta = 0, ..., 7\pi/8$ and the frequency f which depends on the filter kernel's standard deviation σ is $f = 0.01, ..., 0.03$. Zheng et al. [ZLD04] convolve a document image with 16 different Gabor filters having varying θ. Then, they compute the variance between the input element (e.g. character) and the respective Gabor filtered element for each direction. Doing so, text blocks which have a pattern that corresponds to a Gabor filter's frequency and orientation get values close to 1 and -1 while other regions have a σ close to 0.

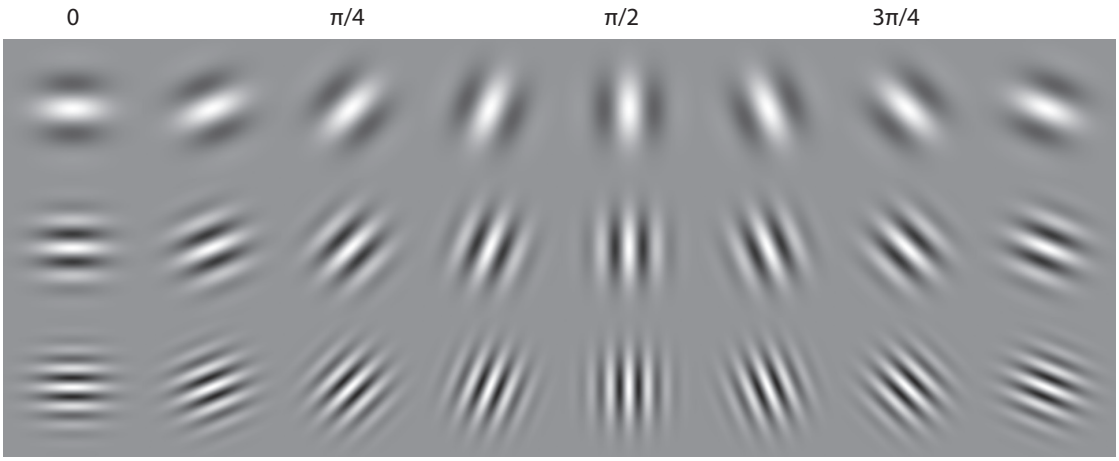


Figure 2.7: Even Gabor filter bank. Columns represent changing angles θ while rows represent increasing frequencies (starting at $f = 0.01$).

2.2.9 Local Descriptors

In contrast to other features outlined in this section, local descriptors are features that capture local structures in an image. The descriptors are either located using an interest point detection (e.g. DoG) or at explicitly defined locations such as sample points at contours. Both local descriptors presented were introduced for general object recognition tasks and later applied to text classification.

SC: Shape Context (SC) is proposed by Belongie et al. [BMP01; BMP02] for handwritten digit recognition and general categorization tasks. In order to reduce the computational complexity of SC descriptors, points of a CC are sampled along the contour with a fixed distance. Then, for a sample point p_i, all relative distances $(q_j - p_i)$ to the remaining sample points q_j are computed. These points are accumulated to a log-polar histogram which is centered at the currently observed sample point p_i. According to Belongie et al. the log-polar histogram accounts for the uncertainty that increases for shapes which are further away from the point p_i.

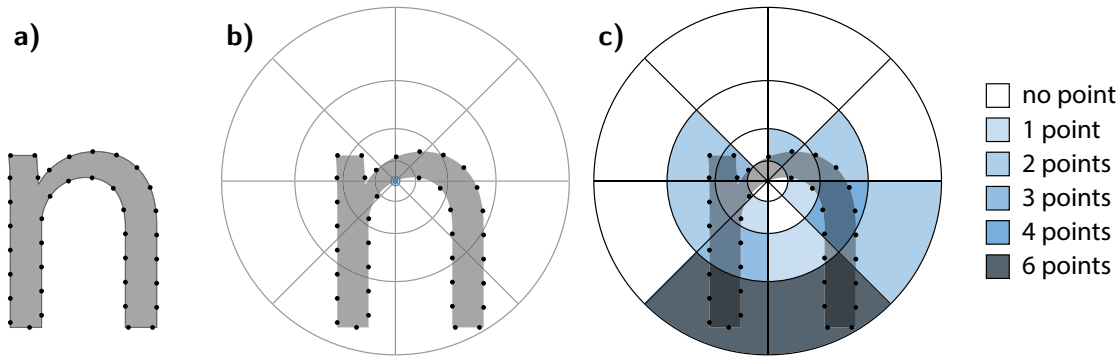


Figure 2.8: Shape context of a sample point (blue point in b)).

Figure 2.8 illustrates the computation of a shape context histogram for a point p_i. First the contour pixels are sampled in a). In this case a coarse sampling is chosen for an improved illustration. A log-polar grid is centered at the currently observed point which is highlighted in b). Note that the grid's radius corresponds to the maximal relative distance of the point set which renders the descriptor scale invariant. For illustration reasons, a coarse grid with eight orientation and four radial bins (= 32 dimensions) is chosen. Finally, the point density is computed for each histogram bin c).

SIFT: A local descriptor frequently used in Computer Vision and Pattern Recognition is the Scale Invariant Feature Transform (SIFT) which is introduced by Lowe [Low99; Low04]. SIFT features are generally extracted at local extrema in a DoG scale-space. By these means, the feature's position and scale are detected. The gradient vectors are then accumulated to 4×4 spatial histograms. Depending on the gradient vector's orientation, they get accumulated into one of eight different orientation histograms. In order to

reflect the edge strength, the gradient magnitude is accumulated. Rotation invariance is achieved by rotating each feature with respect to the local patch's dominant orientation. Figure 2.9 shows the creation of a SIFT descriptor. First the gradient orientation $\theta(x, y)$ and the gradient magnitude $m(x, y)$ are computed. Then, a 128 dimensional descriptor having eight 4×4 orientation histograms is created. Finally, each pixel (highlighted in Figure 2.9) is accumulated with respect to its location and gradient orientation. Note that both the orientation and the gradient magnitude are interpolated in order to increase the descriptor's robustness with respect to slight changes in position.

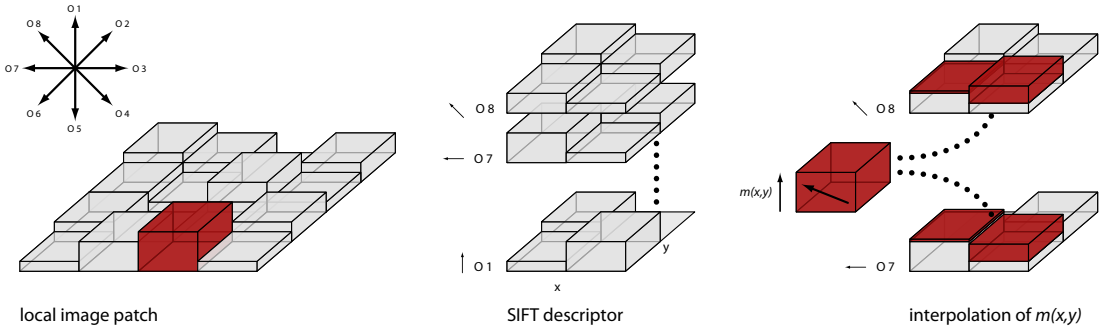


Figure 2.9: Computation of a SIFT descriptor.

Initially, SIFT is proposed for image matching however it was then applied to different object recognition and image processing tasks [Zag+12; DS10]. A successful extension to SIFT is the introduction of BoW. It is proposed for natural scene classification by Li and Perona [LP05]. In order to generate BoWs, an unsupervised training stage collects local image patches or SIFT descriptors from a training set. These features are clustered in feature space. During recognition, each feature extracted gets accumulated to a histogram bin that represents the closest cluster center. By these means, the discriminative power of local descriptors is increased while preserving their robustness with respect to local non-rigid transformations. Zagoris et al. [Zag+12] were the first who applied this approach for text classification.

2.3 Machine Learning

In this section machine learning algorithms for text classification are detailed. It is not an exhaustive description of classifiers since concepts such as Random Forests, Bayes Networks or Conditional Random Fields are not covered. The intention is rather to reflect and summarize commonly used classifiers in the context of text classification. Table 2.2 gives an overview of relevant publications and their respective classifier choice.

2.3.1 Linear Discriminant Analysis

The Fisher classifier [Fis36] was the first classification attempt in pattern recognition. If both classes have a Gaussian distribution, the Fisher classifier finds the optimal solution by maximizing the ratio of the between class-scatter to the within class-scatter. Although the Fisher classifier can be extended to a multi-class classification problem, Zheng et al. [ZLD04] propose to use a one-against-one classification scheme with three classifiers in total. The LDA is closely related to the Fisher classifier with the difference that the covariances are assumed to be equal. The LDA is later used by Kavallieratou et al. [KSA04] to classify handwritten and printed Latin and Greek scripts. Figure 2.10 a) shows the LDA for a two class classification problem. Note that both class distributions are Gaussians. In contrast to a simple SVM LDA is capable of dealing with scattered data even if the class distributions overlap.

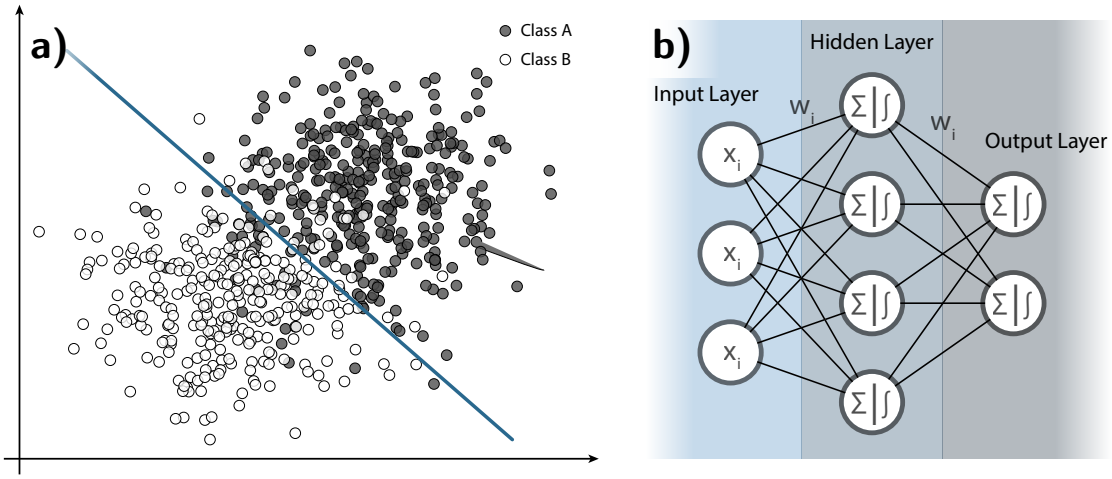


Figure 2.10: LDA of a two class problem a). An NN with one hidden layer in b).

2.3.2 Neural Networks

Artificial Neural Networks (NN) are introduced by Mc Culloch and Pitts [CP43] and composed of multiple neurons. A neuron is an activation function which can be linear or non-linear. Neurons are connected amongst each other so that the output of one layer is fed to all neurons of the next layer. If an NN with one hidden layer as illustrated in Figure 2.10 b) is considered, the first layer feeds extracted features to the NN. Imade et al. [ITW93] use one neuron per feature dimension. The hidden layer has activation functions that are selected by the designer. In case of Multi-Layer Perceptron (MLP) nonlinear activation functions such as Sigmoid functions are used. The output layer – in case of classification – maps the class label. Back propagation is a method for training NNs where the connection weights w_i are updated such that the error between the output

and the expected output is minimized. NNs are used for text classification by Imade et al. [ITW93], Kuhnke et al. [KSK95], and Koyama et al. [KHK08].

2.3.3 Support Vector Machines

Support Vector Machines SVM are first introduced by Vapnik and Chervonenkis [VC74; Vap82]. In contrast to preceding machine learning algorithms such as Perceptrons or NNs, the Vapnik-Chervonenkis (VC) theory does not find any solution by error minimization on the training set, but it tries to find the *best* solution. Hence, statistical learning theory considers the difference between the empirical risk and the true overall risk by incorporating the size of the training data (curse of dimensionality) and the model complexity. The *best* solution for a given classification task is defined as the hyper plane which has a maximum margin $1/||w||$ between the classes. This optimization problem is convex and has therefore one optimal solution which can be efficiently solved using Lagrange Multipliers. Support Vectors are those data points that are located on the boundary (i.e. whose distance to the hyperplane is $1/||w||$) which are generally a small fraction from the training set [BGV92].

A drawback of this learning theory is that solely linearly separable data can be trained and its sensitivity to outliers in the data. Outliers are only crucial if they are located close to the hyperplane. In 1992 Boser et al. [BGV92] extend the SVMs to general non-linear classifiers. They make use of the kernel trick [ABR64] so that the mapping to a higher dimensional space can be computed implicitly. The decision function $D(x)$ for these non-linear kernel machines is:

$$D(x) = \sum_k y_k \alpha_k^* K(x_k, x) + b, \quad \alpha_k^* \geq 0 \tag{2.7}$$

where α_k^* are the Lagrange multipliers, $K(x_k, x)$ is a kernel function, x_k are the feature vectors and $y_k = 1$ for features of class A and -1 for features of class B. In Equation 2.7 solely Support Vectors have a non-zero weight. In [BGV92] polynomial and Radial Basis Function (RBF) kernels where proposed for $K(x_k, x)$. For polynomial kernels, the designer has to choose the polynomial's degree. The RBF kernel function is defined as:

$$K(x_k, x) = e^{-\gamma \|x_k - x\|^2} \quad \gamma \geq 0 \tag{2.8}$$

It can be seen that γ is the only parameter which needs to be determined. In general, this parameter is found by applying a cross-validation on the training set with varying γ. Then, the γ is chosen to be the value which maximizes the classification performance on the training set.

Non-linear kernel functions allow for classifying data that is not linearly separable (e.g. XOR). However, classifying interleaved data would either result in a complex classification boundary which affects the generalization performance of a SVM or for some scenarios it is impossible to find a hypersurface at all. That is why Cortes and Vapnik [CV95] introduce slack variables. The so called soft margin finds a minimum set of *training errors* which are removed when computing the hyperplane. In order to control

the number of training errors allowed a cost constant C is introduced. Similar to γ the cost can be optimized by performing a cross-validation on the training set. In general, a grid is constructed which optimizes γ and C at the same time.

Figure 2.11 illustrates a SVM for a two class problem. White circles indicate features of class A while gray circles indicate features of class B. Note that solely three Support Vectors are needed for the classification task in a). The outlier is ignored if a soft margin is applied rather than the initially proposed margin. In Figure 2.11 b) an RBF kernel is illustrated which separates data non-linearly.

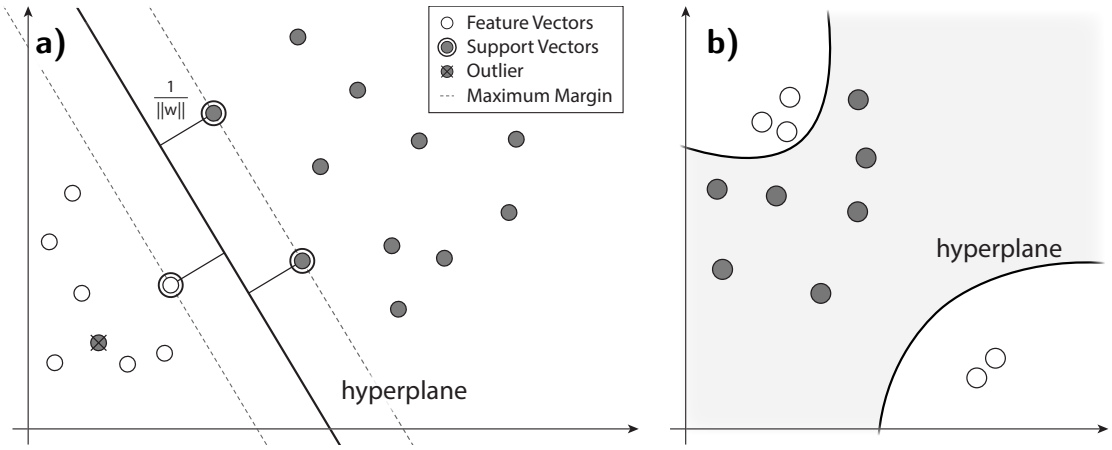


Figure 2.11: SVM with maximal margin hyperplane a). RBF kernel for a non-linear classification problem in b).

For text classification, SVMs are used amongst others by Kandan et al. [Kan+07], Chanda et al. [CFP10] and Zagoris et al. [Zag+12].

2.4 Text Classification

The term *Text Classification* refers to the task of separating machine printed text from handwritten text. Similar tasks in Document Analysis, which are not discussed in this section, include font recognition [JSS99], writer identification [FS13] and text detection in natural scene images [YT11]. The systems proposed for text classification have different designs depending on the data (see Figure 2.12), the script or the objective. The motivations for text classification found in the literature are listed below:

- Pre-processing for OCR [KSK95; FWT98; PC99; GM01; ZLD04; KSA04; FSG06; Kan+07; KHK08]
- Pre-processing for writer identification and signature verification [Pen+09]
- Document retrieval [Pen+09]
- Document segmentation [ZLD04]

- Zone identification for method selection (e.g. Ocr, image understanding) [CFP10]
- Handwriting extraction for form processing [PB11]
- Image compression [ITW93]

Up to now, Ocr and handwriting recognition are fundamentally different due to the different sources. That is why a text classification is performed in advance. Then, the appropriate recognition engine is applied to handwriting and machine printed text respectively. Regarding the previously mentioned motivations for text classification, it can be concluded that in general text classification aims at simplifying subsequent document analysis tasks as specialized algorithms can be applied once handwritten text can be told apart from printed text. An interesting study published by Wamain et al. [Wam+12] shows that humans use different brain cortices when recognizing printed or handwritten text too.

State-of-the-art text classification systems, which are detailed subsequently, are categorized with respect to their design. First *Basic Systems* which utilize document segmentation, feature extraction and classification are depicted. In Section 2.4.2 text classification systems with a post-processing step are mentioned. Then, segmentation free systems are discussed in Section 2.4.3.

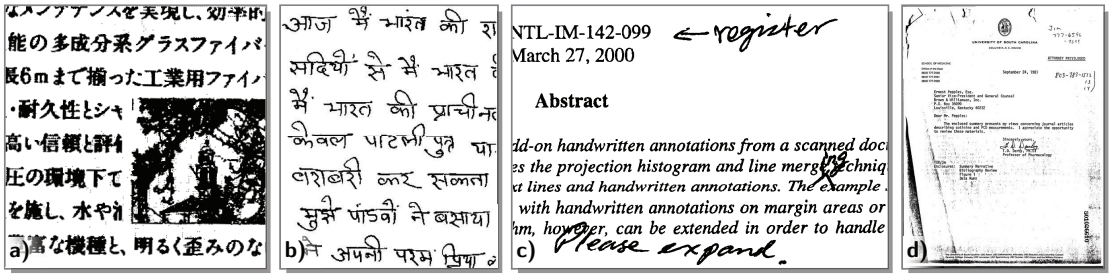


Figure 2.12: Document samples from text classification approaches between 1993 and 2003. Chinese script with images in **a)** courtesy by [ITW93], Bangla script **b)** courtesy by [PC01], English script with handwritten annotations in **c)** courtesy by [GM01] and English script with noisy background in **d)** courtesy by [ZLD04].

2.4.1 Basic Systems

Figure 2.13 illustrates the design of text classification algorithms which are described in this section. All these systems have in common that text (e.g. words or lines) are segmented prior to classification. Depending on the data or the chosen features a preprocessing stage including e.g. skew estimation is applied in advance. Having detected text elements, features are extracted that condense the information while still being distinctive enough to distinguish between printed and handwritten text. These features are classified using a supervised machine learning algorithm (see Section 2.3) or dynamic thresholds.

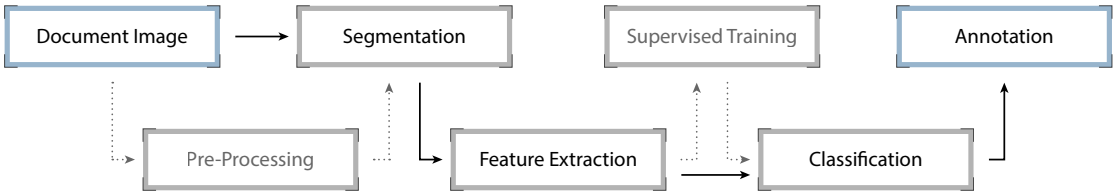


Figure 2.13: Design of a basic text classification system.

An early text classification approach is proposed by Imade et al. [ITW93] that aims at image compression. Therefore, they separate a document into four different classes: *printed characters, handwritten characters, photographs,* and *painted images.* First Imade et al. remove noise by spatial filtering using 8×8 pixel blocks. These blocks are set to zero (foreground) if at least one foreground pixel is present. Based on a-priori assumptions, they filter blobs if at least 16 proximate background elements are detected either vertically or horizontally. Then, the Imade features (see Section 2.2) are computed for randomly sampled 32×32 pixel blocks. The features are a combination of luminance and gradient vector features. These features are classified using a layered NN (see Section 2.3) with a single hidden layer. The NN's input layer consists of 56 neurons which correspond to the feature vector's dimensionality. In order to increase the NNs flexibility, the output layer is designed with five units corresponding to the four classes and an additional background class. All three layers - including the hidden layer with 30 units - are trained with 150 samples (30 per class) using error back propagation [RHW88]. The evaluation is carried out on "1,000 experiments" per class. The overall classification result is 81.98% where *painted images* have the lowest precision being 67.7% and *photographs* perform best with 98.8%.

In 1995 Khunke et al. [KSK95] propose a text classification that aims at supporting OCR systems. They classify text into machine printed and handwritten characters so that specialized OCRs can be used in order to increase the system's performance. Similar to Imade [ITW93] they use a three layer feed-forward NN for the classification task. They state, that features that incorporate *a-priori* knowledge increase the efficiency of a system since the dimensionality is reduced. The features, which were detailed in Section 2.2 are a combination of line straightness and symmetry features. Their training set, which consists of 3,632 character images, is partially taken from the NIST Special Database 3 and scanned by the authors. Khunke et al. evaluate the proposed methodology on a test set containing 1,068 uniformly distributed characters where they reach a classification rate of 78.5%. In contrast to other authors, they consider isolated characters only.

In contrast to this approach, Fan et al. [FWT98] propose a text classification system that incorporates word and character segmentation. Since their segmentation is based on the X-Y cut algorithm they propose a PP skew correction of text blocks. Having segmented text into word or character BBs, CBLV features are computed for each text block. They propose a dynamic threshold which is based on the median BB height

rather than utilizing a classifier. The approach is evaluated on a test set containing 25 handwritten and 25 machine printed text blocks either written in Chinese or English. On this data set a recognition rate of 86% for BBs and 82.5% for text blocks is achieved.

Pal and Chaudhuri [PC99; PC01] present a similar approach to Fan et al. that is applicable to Bangla and Devanagari scripts. They segment text lines – which are their smallest entity – by detecting valleys in the vertical or horizontal PP of a text block. First handwritten text blocks are detected e.g. by thresholding the longest run. For all remaining text blocks CLPSD features are computed which differentiate machine printed and handwritten text lines by computing the standard deviation between profile points and their baseline/lower line. They utilize the dynamic threshold proposed in [FWT98] rather than a classifier. Experiments were carried out on 100 and 600 different documents respectively resulting in an overall accuracy of 98.6%. This approach is later extended by Banerjee and Chaudhuri [BC12] who show that a SVM improves the classification accuracy compared to the dynamic threshold. They achieve a recognition performance of 96%[3] on a data set with 12,830 components collected among Master and PhD students. The best performing classifier evaluated is a SVM having an RGB kernel. Since the RGB kernel is not further discussed, it is assumed that an RBF kernel was used instead. Narayan and Gowda [NG12] pick up the ideas presented by Pal and Chauduri too. They define the PP features in terms of rough sets and evaluate their approach on the IAM-DB.

Jang et al. [JJN04] propose a similar system to Fan et al. that combines handcrafted features (e.g. BB width variance) with layout features (see Section 2.2.2) for text classification of Korean mails. In order to classify the features they train an MLP with one hidden layer.

Another approach that utilizes PPs is proposed by Kavallieratou et al. in [KSA04]. They first segment documents into text lines. In contrast to Guo and Ma, they compute the horizontal PP of a text line's upper and lower profile. Then, three features namely the ratio between the ascender and the main body zone, the ratio between the descender and the main body zone and the ratio between the area and the maximum of the PP are computed for each text line. These features are used to classify text lines into *printed* and *handwritten* using an LDA with the Mahalonobis distance. The approach is evaluated with a 10-fold cross validation on 50 randomly extracted images from the IAM Database [MB99] (IAM-DB) containing English text and the GReek Unconstrained HanDwriting Database [Kav+01] (GRUHD) containing Greek handwriting where they achieve an *overall accuracy* of 98.2%.

Akbarpour et al. [Akb+10] propose a system that differentiates between machine printed and handwritten Farsi. They claim that handwriting is less legible than printed text. In order to measure the *quantity of legibility* Quantity of Sudden Pen (QSP) features are introduced. QSP features are computed by counting a contour's changes of the Chain Code. Then, class labels are assigned based on the minimal distance to the mean of the training set's distribution. Unfortunately, the authors do not mention the robustness of

[3]Note the results cannot be directly compared to those of Pal and Chanduri [PC01] since the data set changed

these features with respect to noise (e.g. white Gaussian noise) which might increase the QSP if the contours are affected.

Another contour based approach for Arabic scripts is proposed by Kumar et al. in [Kum+11]. They first match edges extracted using the Canny edge detector [Can86] with CCs. Then, template lines are matched to the edges. Every line triplet forms a feature. These features are then accumulated to a histogram which is similar to the Bag-of-Words approach. Having extracted the Bag-of-Words for each *zone* – that is defined by Voronoi regions – they are classified into *printed* and *handwritten* text using a SVM. The method is evaluated on 625 images which have clean *zones*. An average pixel level accuracy of 98.2% is achieved on this data set.

Chanda et al. [CFP10] deal with torn documents that are similar to those discussed in this thesis. In contrast to the approach proposed, they suggest a two tier approach. In the first tier Gabor features (see Section 2.2.8) are extracted by means of sliding windows with three different scales. These features allow for discriminating text from non-text (e.g. noise) elements when classified with a SVM having a Gaussian kernel. Then, CCH features are computed for the remaining text elements in the second tier (see Section 2.2.6). Again, a SVM is used for classifying the components into *handwritten* and *printed* text. The method is evaluated on snippets with Latin scripts. In total, 39,190 "text blocks" were classified resulting in an accuracy of 95.94%.

Pinson and Barret [PB11] propose text classification for Latin scripts using Eigenfaces [SK87; TP91]. Therefore, CCs are transformed to fit in a 64×64 window. For each window, the first N and last N Eigenvectors are used as feature representation. The former capture high level character structures while the latter contain detailed structures and edge information. Since the Eigenvectors in between are rejected, the feature dimensionality is reduced to 100. Then, a modified k-NN classifier trained with 87,3010 CCs from both classes (*printed, handwritten*) is used for classification. They propose to take a user selected distance threshold instead of a fixed neighborhood ratio k. If a connected component is classified as *handwritten*, the component is split and further classified in order to reduce the false positive rate. This method achieves a precision of 84.63% on 360 binary images from the NIST SD19 data set. It is mentioned that falsely CCs reduce the method's performance since it relies on perfectly segmented characters. Additionally characters with a simple topology (e.g. l) are likely to be confused with single strokes. The authors do not consider neighboring information to reduce the error rate.

Zemouri and Chibani [ZC11] utilize the Radon transform for discrimination between handwritten and machine printed text. The Radon transform features – outlined in Section 2.2.7 – are extracted for all words that were extracted in a segmentation step. Then, a SVM having an RBF kernel is used for classifying them into *handwritten* and *printed* text. For performance evaluation 21 images were extracted from the IAM-DB where the method achieves a recognition rate of 98.32%. A similar approach based on the Hough transform is later introduced by Konno and Hirose [KH12].

Zagoris et al. [Zag+12] present a text classification methodology that is based on local descriptors. Similar to the approach presented, they first binarize the image and

group CCs by means of an adaptive RLS [Nik+10]. For each thus found component, SIFT features are computed. Then, BOW which capture the descriptors' distribution in the feature space are computed for word components. In contrast to previous work, they propose a classification scheme that implicitly rejects noise elements. In order to do so, two SVMs are trained with a one-against-all scheme. If an element is neither classified as *handwritten* nor *printed*, it is labeled as noise. If an element gets assigned both class labels (i.e. both SVMs assign the label 1), the distance to the closest Support Vector is computed for both kernel machines. Then, the class label of the SVM having the larger distance is assigned. Unfortunately, it is not mentioned how this distance measure is affected by the SVM or kernel parameters (e.g. the cost C). The method is evaluated on 103 images of the IAM-DB and on 100 representative images from the UK Natural History Museum cards where an F-measure of 0.989 and 0.769 is achieved respectively.

2.4.2 Systems with Post-Processing

Systems with post-processing have the same design as basic systems which were previously depicted except for the final relabeling step (see Figure 2.14). In this step a neighborhood between text elements (e.g. words) is established. Based on the assumption of homogeneity, these elements are relabeled if their label is different to those of their neighbors. In contrast to Basic Systems, all systems of this category utilize supervised machine learning approaches for classification.

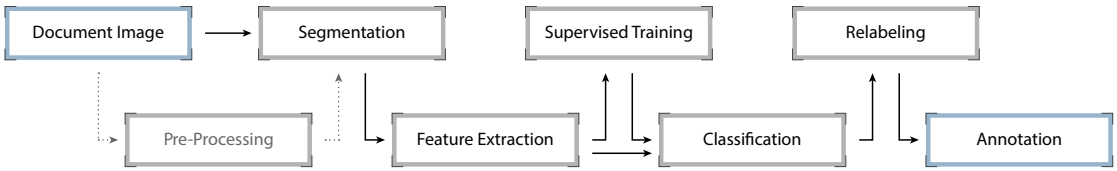


Figure 2.14: Illustration of the design with post-processing that introduce a relabeling stage.

Guo and Ma [GM01] propose a text classification scheme that is applicable for printed documents with handwritten annotations. Note while previous works focused on classifying either text lines or text blocks as printed or handwritten text, they focus on the classification of single bounding boxes. Their segmentation is based on CC analysis of previously binarized documents. In order to group single characters (of printed text) into words, the local geometry is analyzed. Then, normalized PPS are computed for every character BB. These features are fed into two HMMS one of which is trained on printed, the other on handwritten characters. The HMMS are composed of 62 hidden states corresponding to the Latin characters (upper and lower case) and 10 digits. Text elements are then classified according to the maximal *a-posteriori* probability of both HMMS. Finally, potential classification errors are corrected by observing the labels of neighboring elements and relabeling elements that nicely fit other elements with a different class label. For evaluation 25 different documents containing 187 handwritten

words were scanned. The results were solely computed for handwritten words, where this method achieved a recall of 72.2% with a precision of 92.86%.

Zheng et al. [ZLD02; ZLD04] present two methodologies to classify Latin text into machine printed and handwritten zones in noisy documents. Similar to Guo and Ma [GM01], they first extract CCs which are merged to words with respect to their geometric proximity and size. In their first approach [ZLD02] they propose a 108 dimensional feature vector. This vector is the combination of structural features, a bi-level co-occurrence histogram, a bi-level $N \times M$-gram, Pseudo Run Lengths and Gabor Features (see Section 2.2). These features are classified using a Fisher classifier. The test set is a collection of 318 noisy documents where an accuracy of 97.3% is achieved on word level when tested with a 10-fold cross validation (on character level they achieve an accuracy of 93.0%).

In [ZLD04], Zheng et al. extend their previous approach. In this publication they use the same feature set except that Crossing Count Histograms are added. In order to overcome the curse of dimensionality and to speed-up the computation, a feature selection is performed which reduces the feature set from 140 to 31 dimensions. A comparison of three classifiers – namely k-NN, Fisher classifier and SVM – shows that the SVM performed best with an overall accuracy of 96.0% compared to the Fisher Classifier (95.5%). Nevertheless, the authors propose the use of the Fisher Classifier because of its computational speed. In contrast to their prior publication [ZLD02] Zheng et al. introduce a third class (*noise*) beside the *handwritten* and *machine printed* classes. This class allows for rejecting – falsely binarized – foreground elements. Guo and Ma [GM01] were the first to propose a post-processing step that re-labels elements with respect to their neighboring elements. However, Zheng et al. propose a more sophisticated post-processing stage which utilizes an MRF. Therefore, cliques are defined for printed text and noise. The former is applied on printed blocks where solely the nearest neighbor block is connected. For the latter four nearest neighbors are added to the clique of the currently observed noise block. Then, the blocks are re-label until the energy of the corresponding Gibbs distribution is minimized. The evaluation is again carried out on 318 business letters from the tobacco industry litigation archives. In this evaluation scenario 94 images are used for testing while the remaining 224 images are used as training set. The MRF post-processing stage increases the accuracy to an overall accuracy of 98.1%. Interestingly, the *handwriting* class – which is the only class without its predefined clique – is the only class, whose error is increased by 2.1% rather than reduced. This approach is later successfully applied to Arabic scripts by Farooq et al. [FSG06], and Mozaffari and Bahar [MB12]. Belaïd et al. [BSD13] and Saïdani et al. [SEB13] use features proposed by Zheng et al. in combination with a SVM. They use a k-NN voting scheme instead of deploying the MRF.

Kandan et al. [Kan+07] propose a similar architecture as Zheng et al. [ZLD04] for discriminating handwritten from printed document elements. Instead of using three classes – where one class detects noise – they propose a separate pre-processing step where noise is removed. The noise removal is based on geometrical constraints of CCs (e.g. aspect ratio of the BB). They then extract Hu invariant moments (see Section 2.2.4)

despite the findings of Flusser [Flu00]. The moment invariants are classified using a SVM with an RBF kernel. After classifying all CCs of a document, a post-processing step is proposed which relabels the components with respect to their neighbors. The neighbor relations are established using Delaunay triangulation [Dav+96]. For document images, Delaunay triangulation has two drawbacks. First, the relationships established are not related to the reading direction or composition of text. Second, if there is a large gap between components (e.g. end of paragraph and footnote or noise at the bottom of a page), the Delaunay triangulation still connects these components. The latter is compensated by thresholding neighborhood relations based on their distance. Having established a relationship between the CCs, a new class label is assigned to a component if more than 50% of its neighbors having a *similar* height (± 7 px) have another class label. The approach is evaluated on 150 document images that are similar to those presented in [ZLD04]. The classification accuracy is 87.9% with a k-NN classifier and 93.22% with the SVM proposed. Unfortunately, the improvements of the post processing step are not detailed. A similar approach which uses Hu invariants in combination with a k-NN classifier is later proposed by Hangarge et al. [Han+13]. They evaluate their approach on the IAM-DB and a newly introduced database that has a similar structure and contains 100 pages with South Indian scripts. Using a 10 fold cross validation, an average precision of 99.26% is achieved.

Peng et al. [Pen+09; Pen+11] propose a text classification that aims at separating printed text from annotations using an MRF. Therefore, they extract handcrafted features similar to those from Khunke et al. [KSK95], CC features and Gabor features similar to those proposed by Zheng et al. [ZLD04]. Instead of classifying these features directly, they apply a G-means clustering [HE03]. The G-means clustering is a k-Means clustering that estimates k by splitting the data until all clusters have a Gaussian like distribution. The clusters are labeled into *handwritten, printed,* and *noise* in a preceding training stage. The CCs which are labeled by these means, are re-labeled using an MRF similar to the text classification system proposed in [ZLD04]. In contrast to Zeng et al. , the neighborhood is based on a Gaussian-like metric i.e. the distance is defined with respect to the convex hull of a CC. Another difference to the post processing step proposed by Zheng et al. is that Peng et al. do not only consider patch clique occurrence frequency but also incorporate the previously extracted features. Hence, similarities in the MRF are established by means of the features extracted and the geometrical configuration of CCs.

A second post processing step is proposed which separates machine printed from handwritten text if they are overlapping. In this scenario, Peng et al. only consider elements that were previously labeled as *handwritten*. For these components, SC features (see Section 2.2.9) are extracted for every pixel. Then, foreground pixels with "similar" features are aggregated in order to expand the individual labels. Finally, the previously described MRF is exploited so as to establish the final class labels of a CC.

2.4.3 Segmentation Free Systems

Since text segmentation is crucial for the performance of text classification, segmentation free systems were proposed in [ZL12; KHK08]. These systems use sliding windows for feature extraction. In contrast to the previously depicted systems the output is a heat map rather than labeled elements.

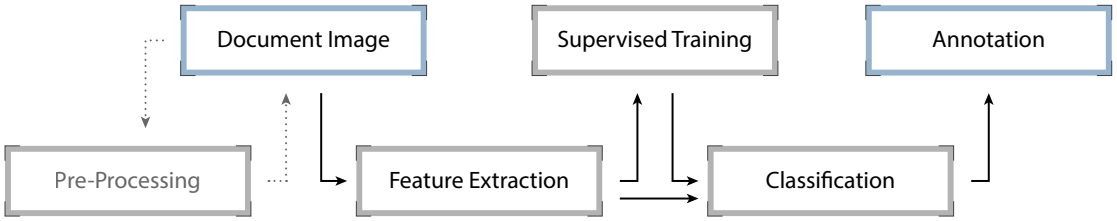


Figure 2.15: Design of segmentation free systems.

Zhang and Lu [ZL12] invented a three step text classification process. The feature extraction is similar to the co-occurrence features presented by Zheng et al. [ZLD04]. However, they extract ECM features (see Section 2.2.5) for overlapping block windows which allows for text classification without the need of segmentation. These blocks are classified with a linear SVM. Then, a relation between blocks is modeled by means of a DRF introduced by Kumer and Hebert [KH06]. In the DRF each node represents a block where the inputs are the predictions of the SVM. A Chinese and a Latin data set (IAM-DB) were used for evaluation. The former consists of 2,652 handwritten and printed characters. For the latter database 50 images were extracted and the evaluation is carried out using a 10 fold cross validation where they achieve a precision of $\approx 99\%$.

A text classification approach that has no need for character segmentation or binarization is proposed by Koyama et al. [KHK08; KKH08]. First, they extract local power spectra by means of a fixed width sliding window. They use the FFT in order to obtain the power spectrum. However, as stated in the paper, every other transform (e.g. DCT, Wavelet transform) can be used as well. Then, they extract so-called SDLFD features (see Section 2.2.7) which are classified using an MLP with one hidden layer. Since no segmentation or binarization was carried out to localize characters, their final output is a *heat map* with classification probabilities which can be thresholded for fixed class labels. Unfortunately, Koyana et al. do not discuss further post processing steps that allow a mapping between the characters and the final classification map. The evaluation is carried out on a subset of the ETL character database containing 4,000 characters with 500 printed and 500 handwritten characters from Chinese, Hiragana, Katakana, and Latin respectively. They reach a *distinction rate* of 97% for extracted characters. If sample documents are classified, the performance seems to drop[4]. However, they published classification maps rather than numerical results for this evaluation.

[4] 3 out of ≈ 48 characters are wrong which results in a precision of 93.8%

2.4.4 Text Classification System Comparison

Though text classification is an ongoing research topic [Han+13; BSD13] there is no appointed benchmarking data set or competition such as DIBCO [PGN13]. Hence, comparing different approaches for text classification has to be treated carefully. In order to account for the scientific impact of the approaches, a citation graph is given in Figure 2.16. Additionally, Table 2.2 summarizes the results of the approaches described in Section 2.4.

Figure 2.16 illustrates citations of scientific papers. Each circle represents up to two papers if they have the same first author and describe a similar approach (e.g. [KHK08; KKH08]). The first author's last name and the publication year are used to identify the publications. The small circle in the upper right corner bunches the citations of the respective paper. Note that Imade et al. [ITW93] and Kuhnke et al. [KSK95] do not have any edges since they were the first who proposed text classification aiming at separating handwritten from printed text. The circle's size and the stroke opacity indicate the scientific impact of a paper. Note that the total count of citations is regarded which weights older publications more. To demonstrate this, one could divide the total citation amount by the number of years a paper is published. In that case Chanda et al. [CFP10] would have a weight of $4/(2013 - 2010) = 1.3$ while Pal et al. [PC01] who has most references in total would have a weight of $17/(2013 - 2001) = 1.41$ which is closer than their current relation (Pal = 1, Chanda = 0.24). Nevertheless, the absolute count is chosen as citations do not exhibit a linear relation with respect to the years published. In brief, though the citation graph illustrates a paper's scientific impact, these precariousness should be kept in mind when drawing conclusions. In addition to the size and weight, colors indicate a paper's category which was previously introduced.

Table 2.2 summarizes the evaluation results of state-of-the-art text classification approaches. Again, solely the first authors are listed for saving space. Fan et al. [FWT98] and Pal et al. [PC01] use a dynamic threshold for labeling rather than a classifier. The size of the database is listed in pages. If the authors do not mention how many pages are used for evaluation, an approximate page number is estimated to allow for a better comparison. A page in this context is assumed to be DIN A4 and have 530 words which is roughly the mean word count for this thesis. For these publications, the original count and unit is denoted in squared brackets. It can be seen in Table 2.2 that most methods were evaluated on a newly created data set which is not publicly available. Since a system's performance is closely related with the quality of the data a subjective rating is added. In this rating system ●●● stands for challenging data while ○○○ is chosen for data that has clearly separated handwritten and printed regions. Figure 2.17 shows three samples with different ratings. The first of which is challenging data with noisy background. The second is rated ●●○ since handwritten annotations overlap with machine printed text. The third shows an example of clearly separated data.

Table 2.2 shows that the system performance generally increased over the years even for challenging data. While the method of Imade et al. [ITW93] had a precision of 81.9% in 1993 Kumar et al. [Kum+11] achieved 98.2% on a challenging data set in 2011. Another interesting fact that can be seen in this table is that the size of the evaluation

Figure 2.16: Illustration of scientific citations. Note that just the first author is mentioned for compactness. The circles are scaled with respect to the number of a paper's citation count.

set did not necessarily increase over time (e.g. Zheng et al. [ZLD04] had 94 pages in 2004 while Zhang et al. [ZL12] had solely 50 pages for evaluation in 2012). This is odd because the computational power and storage did increase. Table 2.2 allows to draw the conclusion that SVMs are the most popular machine learning algorithms used for text classification. Finally, it should be mentioned that there is a need for a publicly available benchmarking database and an international competition so that the methods can be directly compared based on quantitative data.

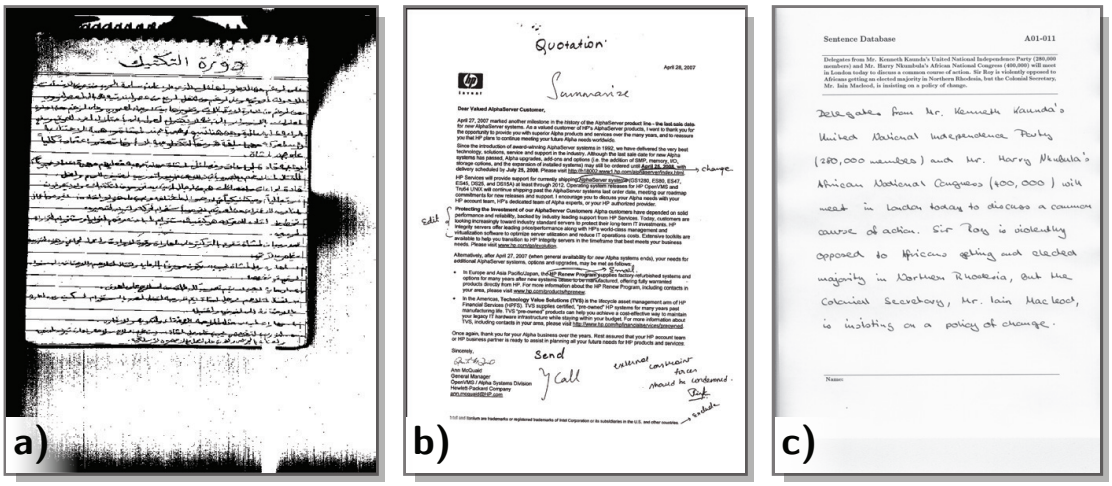


Figure 2.17: Sample pages having different ratings. a) challenging data (●●●) courtesy by Kumke et al. [Kum+11], b) text with handwritten annotations (●●○) courtesy by Peng et al. [Pen+11] and an example page from the IAM-DB in c) which is rated ○○○.

Table 2.3 shows the scripts which are analyzed in the respective publications. It can be seen, that Latin is the most popular script for text classification. Already in 1998 when the research topic text classification was not as popular as nowadays, Fan et al. [FWT98] investigated the classification of Chinese characters. Hangarge et al. [Han+13] explored recently a text classification system that is applicable for four different South Indian scripts and Latin.

2.5 Document Layout Analysis & Page Segmentation

Document Layout Analysis and *Page Segmentation* are interchangeably used in the literature [Ant+09b]. They both refer to the process of transforming an image pixel representation into a higher level representation by forming groups of text lines, text blocks and/or graphical elements. The subsequently enumerated applications are reported for layout analysis:

- Pre-processing for OCR [WCW82; FHD90]
- Document classification [Esp+90]
- Document annotation (for further processing) [OGo93]
- Information retrieval [EDC97]
- Efficient document storage [EDC97]

Author	Classifier	data set	Size		Precision
Imade [ITW93]	NN	own	??	●●●	82%
Kuhnke [KSK95]	NN	NIST	≈ 1/3 page [1,068 chars]	●○○	78.5%
Fan [FWT98]	–	own	≈ 12 pages [50 text blocks]	●○○	86%
Pal [PC01]	–	own	≈ 150 pages [600 images]	●○○	98.6%
Guo [GM01]	HMM	own	25 pages	●●○	92.9%
Zheng [ZLD04]	k-NN	own	94 pages	●●●	94.2%
	Fisher	own	94 pages	●●●	95.5%
	SVM	own	94 pages	●●●	96%
Kavallieratou [KSA04]	LDA	IAM-DB GRUHD	50 pages	●○○	98.2%
Kandan [Kan+07]	k-NN	own	150 pages	●●●	87.9%
	SVM	own	150 pages	●●●	93.2%
Koyama [KHK08]	MLP	ETL	≈ 1 page [1,000 chars]	○○○	97%
Chanda [CFP10]	SVM	own	≈ 73 pages [39,190 words]	●●○	95.9%
Kumar [Kum+11]	SVM	own	625 pages	●●●	98.2%
Pinson [PB11]	k-NN	NIST	360 pages	●●○	84.6%
Zemouri [ZC11]	SVM	IAM-DB	21 pages	○○○	98.3%
Peng [Pen+11]	MRF	own	≈ 55 pages [110 patches]	●●○	86.8%
Banerjee [BC12]	SVM	own	≈ 24 pages [12,830 CCs]	●○○	96%
Zagoris [Zag+12]	SVM	IAM-DB	103 pages	○○○	98.9%
		own	100 pages	●●○	76.9%
Zhang [ZL12]	SVM	IAM-DB, own	50 pages	○○○	99.9%

Table 2.2: Summary of state-of-the-art evaluations and databases. The dots (e.g. ●○○) are a subjective rating system that indicates how challenging the test set is.

The layout analysis approaches are grouped according to the data they are designed for. Figure 2.18 shows three different layout types. In a) a *Manhattan* layout is shown. Note that all elements have borders that are parallel or perpendicular to each other [KSI98]. An example page having *non*-Manhattan layout is illustrated in b). Here, elements such as the illustration have an arbitrary shape. Both, Manhattan and non-Manhattan layouts are referred to as non-overlapping since different elements are separated by white space. An overlapping layout is shown in c). The layout analysis systems can be divided into bottom-up approaches such as the area Voronoi which merge small elements and

Script	Language	Systems
Latin	English, German, ...	[ITW93; KSK95; FWT98; GM01; ZLD04; KSA04; Kan+07; KHK08; CFP10; Pen+11; PB11; ZC11; KH12; NG12; Zag+12; ZL12; BSD13; Han+13]
Chinese	Chinese	[FWT98; KHK08; ZL12]
Bangla	Bengali (Indian)	[PC01; BC12]
Hiragana	Japanese	[KHK08; KH12]
Katakana	Japanese	[KHK08; KH12]
Arabic	Arabic, Persian, ...	[Kum+11; MB12]
Devanagari	Sanskrit (Indian)	[PC01]
Hangul	Korean	[JJN04]
Greek	Greek	[KSA04]
Farsi	Persian	[Akb+10]
Kanji	Japanese	[KH12]
Kannada	Kannada (Indian)	[Han+13]
Telugu	Telugu (Indian)	[Han+13]
Tamil	Tamil (Indian)	[Han+13]
Malayalam	Malayalam (Indian)	[Han+13]

Table 2.3: Different scripts that are the object of investigation for the respective publications.

top-down approaches such as the X-Y cut algorithm which split documents until a certain stopping criteria is met.

2.5.1 Manhattan Layout

The first approaches in document layout analysis (e.g. [WCW82; NS84]) were designed for documents with Manhattan layouts. On the one hand this can be attributed to the fact that it is easier to deal with Manhattan layouts. But on the other hand the number of documents with complex non-Manhattan layouts started increasing in the 90's according to Kise et al. [KSI98].

Wong et al. [WCW82] present an exhaustive document analysis system. Page segmentation, which in their case aims at component grouping for text recognition, is realized by combining a horizontal RLS with a vertical one (see Figure 2.19). The logical AND operation of the thus smoothed images guarantees that columns are not merged. The components are labeled as text or graphics using area and aspect ratio thresholds. This approach requires a skew correction and a constant character or line spacing.

Figure 2.19 shows the RLS for a document image of an article. For illustration a global binarization is chosen where all pixel smaller than one are set to zero. Then, the horizontal and vertical RLS are computed with a length of $l_h = 40, l_v = 140$. Note that the vertical run length is generally larger since it needs to gap text lines rather than

Figure 2.18: Three different layouts that are commonly dealt with in the literature. Manhattan layout in a), Non-Manhattan layout of the PRImA database [Ant+09a] in b) and a handwritten medieval document of the *Saint Gall* database [Fis+11] in c).

white spaces. The final segmentation result can be seen in e) which is a combination of the smoothed images. This approach is extended by Fisher et al. [FHD90] who use a dynamic threshold that is based on the interline spacing which is found in the Hough domain. The dynamic threshold allows for adopting to text line changes within different document images. Okamoto and Takahashi [OT93] also adopted the RLS page segmentation approach. They additionally introduce black separators which detect long vertical lines that are used for column separations in some layouts.

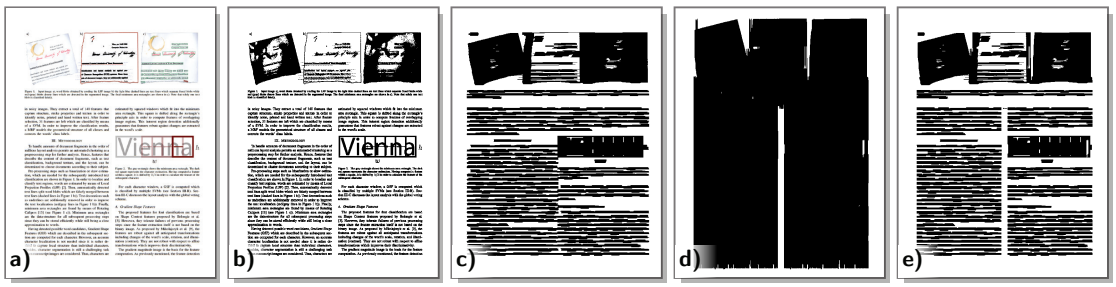


Figure 2.19: Document image a), binary image b), resulting image of horizontal c), and vertical d) RLS. Combination of both RLS images with a logical AND e).

In 1984 Nagy and Seth [NS84] presented the recursive X-Y cut algorithm which is a top-down approach for page segmentation. In this approach the page is recursively divided by e.g. analyzing the PPS of rectangles and splitting these with respect to the largest valleys. Wieser and Pinz [WP93] later combined the recursive X-Y cut algorithm with the RLS in order to extract the layout of newspaper articles.

The motivation of Pavlidis et al. [PZ91] is similar to that of [WCW82]. Both try to

detect text regions in printed articles for further processing such as text recognition. In contrast to previous work, Pavlidis et al. focus on background rather than foreground elements. Long background runs are again found by means of RLS. Text blocks are extracted by combining so called column intervals which are long white runs (areas with continuous background).

Ittner and Baird [IB93] propose a layout analysis by first detecting white runs. They argue that white runs are easier to detect than the – in some examples – challenging foreground elements. White runs are detected by greedily unifying elongated rectangles that are located on the background. The thus disconnected regions are identified as text blocks. They do not mention in detail how the unifying of rectangles is computed.

Etemad et al. [EDC97] apply Wavelet Packets for page segmentation. The advantage of Wavelet Packets to Gabor filters is their computational speed and the implicitly computed multi scale representation. In order to segment a printed page which has Manhattan layout into *text, graphics,* and *background* regions, a soft decision vector is computed for each Wavelet Packet window. The soft decision vectors account for local uncertainty in windows that have e.g. mixed graphical and textual content. They are computed by means of NNs. For the final hard decision criterion, the soft decision vectors of multiple scales are combined such that the class label of the closest class candidates gets assigned.

2.5.2 Non-Manhattan Layout

O'Gorman [OGo93] attempts layout analysis by the so-called docstrum. He focuses on printed document pages with all graphical elements removed. In order to compute the docstrum a k-NN is applied to the centroid of CCs. He chooses k to be five so that the docstrum does not only account for character spacing but text line spacing at the same time. Figure 2.20 b) shows the k-NN applied to a detail of the document in a). Plotting all connections results in the docstrum which can be seen in c). This plot allows to directly detect a document's skew, character spacing and text line spacing. Text lines are computed by tracing the nearest neighbor of characters until they exceed a distance threshold which was previously found in the docstrum. In order to locate the text lines more accurately a regression is performed on all centroids which belong to a single text line. Then, text blocks are formed if lines are approximately parallel and again do not exceed a distance threshold. This method performs well if similar fonts are present. However, large text elements such as headlines cannot be extracted due to the global distance threshold.

The idea of O'Gorman is adopted by Simon et al. [SPJ97]. Instead of computing the Euclidean distance between the centroids of CCs they propose a distance measure that speeds up the k-NN. Their distance is computed by the outer distance of BBs that are fit to the CCs i.e. the difference of the closer edges of two bounding boxes. Words and text lines are grouped by means of locally adopted distance thresholds. Components need to have a vertical overlap of more than 70% of their height in order to be grouped into the same text line.

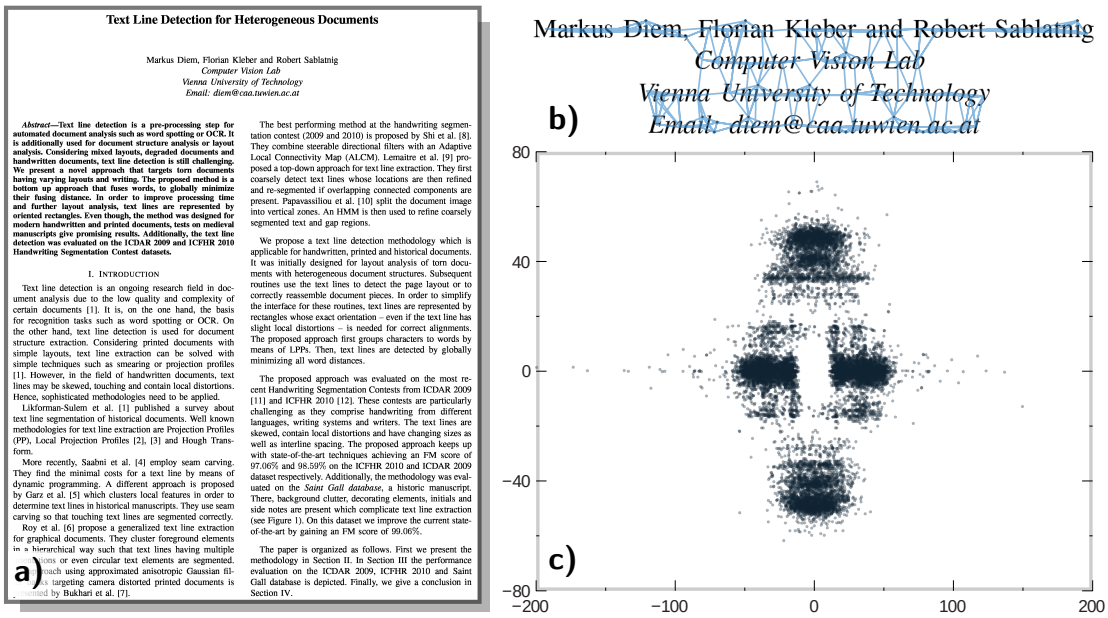


Figure 2.20: Sample document page a), k-NN of a detail b) Docstrum plot of the document c).

Jain and Zhong [JZ96] propose a texture based layout analysis system. As detailed in Section 2.4 the document's components are first classified into *text, line drawing, picture,* and background by means of gradient filters and NNs. Then, text blocks are extracted by simply estimating the BBs of the respective components.

Kise et al. [KSI98] address page segmentation for non-Manhattan printed pages. They therefore extract the area Voronoi diagram on the document's CCs. Now, page segmentation can be interpreted as removing Voronoi edges correctly. Kise et al. propose to delete edges either if two Voronoi regions have a similar area or their distance is below a certain threshold. This approach is later combined with the Docstrum to overcome drawbacks caused by diacritics [AD09].

Xiao and Yan [XY03] adopt the Voronoi diagram for text region extraction. Again, printed documents with non-Manhattan layouts are considered. Instead of computing the Voronoi diagram of CCs, Xiao and Yan propose using Delaunay triangles which is the dual graph representation of a Voronoi diagram. Text regions are then extracted by statistical computations on the Delaunay triangles which consider the length of the triangle's edges (i.e. distance between elements) and the edge orientation.

Ray Smith [Smi09] presents a layout analysis system based on tab stop analysis. Therefore, CCs are filtered such that solely tab stop CCs are left. Then, text lines are formed by connecting left and right tab stop CCs and grouping those on the connecting line. Finally, columns are found by fitting lines to the left and right tab stop candidate CCs. A similar thread is discussed in [CYL13]. In contrast to Smith, they detect white spaces by connecting the CCs with a so-called white space rectangle. Then, vertical white

space rectangles are filtered such that solely those are left which lie between columns. This method won the ICDAR 2013 Competition on Historical Newspaper Layout Analysis.

2.5.3 Handwritten Layout Analysis

Layout analysis of handwritten documents has gained little attention [BI11; GSD11; Mon+09; Mon+10] in the document analysis community. This can be traced back to the fact that modern handwritten documents are mostly unstructured. Hence, researchers focus rather on text line segmentation (see Section 2.6).

Baechler and Ingold [BI11] perform layout analysis of medieval handwritten documents namely the St. Gall Database. In contrast to previous approaches, they propose the use of a multi-scale model. Features such as pixel position, the color value, neighboring RGB values and the MLP output of finer scales are computed. These features are then classified using a dynamic MLP. This approach is extended by Wei et al. [Wei+13]. They evaluate it on two additional historical databases and compare different classification schemes including SVMs and GMMs. In this paper it is shown, that SVMs have a similar performance as the dynamic MLPs while GMMs have a significantly worse performance. Unfortunately, they evaluate pixel errors rather than geometrical errors which does not account for errors such as merging two columns.

Garz et al. [GSD11] extract layout information from medieval Glagolitic manuscripts. Since binarization is challenging in the presence of noise and bleed through, they propose layout analysis on the basis of local descriptors. These are classified into *text* and *decorative elements* using a SVM and subsequently merged on the basis of the class voting and their respective region.

Montreuil et al. [Mon+09; Mon+10] propose a layout analysis of modern handwritten documents such as letters. They therefore exploit hierarchical CRFs which iteratively segment documents into CCs, words, lines, and text blocks. For the final block labeling, features such as relative position of the centroid to the text line or number of words are computed.

2.5.4 Evaluation of Layout Analysis Algorithms

In contrast to text classification there has been a considerable effort in benchmarking layout analysis systems since the beginning of this century [GMA01]. In this chapter a brief overview of the development is given. In 2001 Gatos et al. [GMA01] organized the first Page Segmentation Competition. They evaluated three algorithms on 20 newspaper images from both Greek and English newspapers. The consecutive competition [AGK03] is evaluated on a database with a higher variety including technical articles, memos, faxes, magazines, and advertisements. In this competition again, three participants were evaluated on 32 document pages. In the ICDAR 2005 Page Segmentation Competition the ground truth regions were extended [ABG05]. For the ICDAR 2009 Page Segmentation Competition a new performance metric is introduced that differentiates between merge and split errors that destroy the reading order of a document and those who do not [Ant+09b]. In addition state-of-the-art OCR systems – namely

ABBYY FineReader and OCRopus) were compared to the submitted methods and the XML ground truth representation (Page Analysis and Groundtruth Elements (Page)) is standardized [Ant+09b; CPA11a]. Since 2011 historical documents from the Impact database are used. In addition to the region based error metrics, an Ocr metric is introduced that measures the Ocr error when changing the layout analysis method [CPA11b]. Figure 2.21 shows some sample images of the respective competitions. Note that the images were binarized in order to reduce pre-processing effects until the Icdar 2009 Page Segmentation Competition. Table 2.4 gives an overview of the changes of the respective competitions. It can be seen that the number of participants and the dataset size slightly increased over the years. In addition the papers of the winners are referenced in case their method is published.

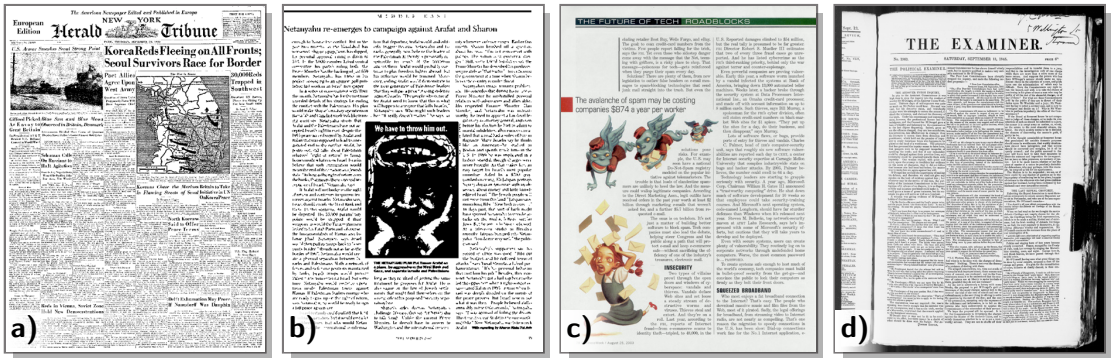


Figure 2.21: Icdar 2001 Newspaper Segmentation Contest [GMA01] a), Icdar 2003 Page Segmentation Competition [AGK03] b), 2009 [Ant+09a] c) and 2011 [Ant+11] d).

In addition to the biennial layout analysis competitions, there have been efforts in comparing existing layout analysis systems. Mao and Kanungo [MK01] compare the X-Y cut [NS84], the Docstrum [OGo93] and the Voronoi Diagram [KSI98] algorithms with two commercial products. They evaluate the methods on 978 document pages collected at the University of Washington [MK01]. The Docstrum and Voronoi Diagram have a similarly good performance while the X-Y cut has the most split and merge errors. It should also be mentioned that the Voronoi Diagram algorithm is on average significantly (2×) faster than the Docstrum [MK01]. A new error measure based on level sets is introduced by Mao and Kanungo [MK01] that detects split and merge errors. However, Shafait et al. [SKB08] show that this measure detects no error at all if the whole page is segmented as one block. In addition to the methods compared in [MK01], they compare the Rls algorithm [WCW82] and the white space analysis [IB93; Bai94]. Their dataset consists of 1,600 articles having Manhattan layouts. Interestingly, their Docstrum implementation is faster than the Voronoi Diagram algorithm. Though their dataset is extended and the error measurement is improved so that the drawbacks of the level set method were cleared, the same methods (Docstrum and Voronoi) performed best. But it is pointed out that both methods have a lack of robustness with respect to changes in font size.

	#	Dataset	Metric	#	Winner
ICDAR 2001	20	newspaper	NSM	3	Liu et al. [Liu+01]
ICDAR 2003	32	articles, memos, faxes, magazines	EDM	3	Goey [AGK03]
ICDAR 2005	26	articles, magazines	EDM	4	Chowdhury et al. [Cho+07]
ICDAR 2007	32	articles, magazines	EDM	3	Chowdhury et al. [Cho+07; AGB07]
ICDAR 2009	55	articles, magazines	PRImA	6	Konya et al. [Ant+09b]
ICDAR 2011	100	historical documents	PRImA, OCR	5	Fontain [Ant+11]
ICDAR 2013	50	historical documents	PRImA, OCR	7	Chen et al. [CYL13]

Table 2.4: Competition overview. Number of test images, modality, performance metric, number of participants and the respective winner.

Recently, Deryagin [Der13] introduces a new page segmentation error metrics that focuses on page segmentation for OCR driven scenarios. In contrast to previous metrics [Ant+09a; MK01; SKB08], his metric is defined such that over-segmentation in reading direction is allowable since modern OCR systems can deal with such kind of errors. It can be seen from the various evaluation methodologies presented, that the definition of ground truth and the definition of errors for layout analysis systems is ambivalent depending on the data and scenario.

2.6 Text Line Segmentation

In contrast to Layout Analysis approaches which generally incorporate strong assumptions about the structure of a document (e.g. [YW13]), text line segmentation methods have no need for assumptions other than the presence of text. These techniques are basically applied to unstructured handwritten documents. Text line segmentation is generally applied for further document processing such as structure extraction, OCR, handwriting recognition or word spotting [LZT07]. Sulem et al. [LZT07] give a survey about text line segmentation for historical documents.

In this section, we first discuss two historical approaches which were introduced for printed documents. Then, smearing techniques for text line segmentation are presented in Section 2.6.1. Approaches which segment text lines using graphs are discussed in Section 2.6.2 and finally other techniques are presented in Section 2.6.3.

An early approach for text string separation is presented by Fletcher and Kasturi [FK88]. They focus on printed text in technical drawings. In order to extract text strings of arbitrary size, they first binarize the images, extract CCs and filter them with respect to an area and an aspect ratio threshold. CCs which are co-linear and comply with a local height threshold are clustered together. The clustering is carried out in the Hough space to improve the performance. These text strings are further subdivided into

words by thresholding the local distance of CCs. A similar approach to this is proposed by Hönes and Lichter [HL93]. They also extract text lines of arbitrary orientation in machine printed documents. In contrast to Fletcher and Kasturi they threshold the angle between consecutive gradient vectors which are connecting the center of mass of CCs.

2.6.1 Smearing Techniques

Techniques which are based on smearing first transform the input image to a smeared image by means of LPPs, Steerable filters or anisotropic Gaussians. The filtered image is then assumed to resemble the text line's PDFs which are further processed. In general, smearing – if its parameters are estimated poorly – is sensitive to global or local skew changes. However, Ziaratban and Faez [ZF11] and Bukhari et al. [BSB11b] propose smearing techniques that can cope with skewed text lines. Figure 2.22 shows two typical smearing techniques. The document snippet is a sample from the CVL-DB [Kle+13] which is rotated by 15° in b). The upper row illustrates anisotropic Gaussian filtering of the gray-scale image and the thresholded text line PDFs. For the Gaussian filter $\sigma_x = 58$ and $\sigma_y = 12$ are chosen. For comparison, the images are filtered with an LPP in the lower row (the filter kernel here is chosen to be 350×1 px). It can be seen that the Gaussian filter is more robust with respect to skew changes. The advantage of LPP filters is speed since solely 2 values need to be updated per pixel if Dynamic Programming is used. Anisotropic Gaussian filters are used in e.g. [Li+08] while [Yos+09] extract text lines using LPPs.

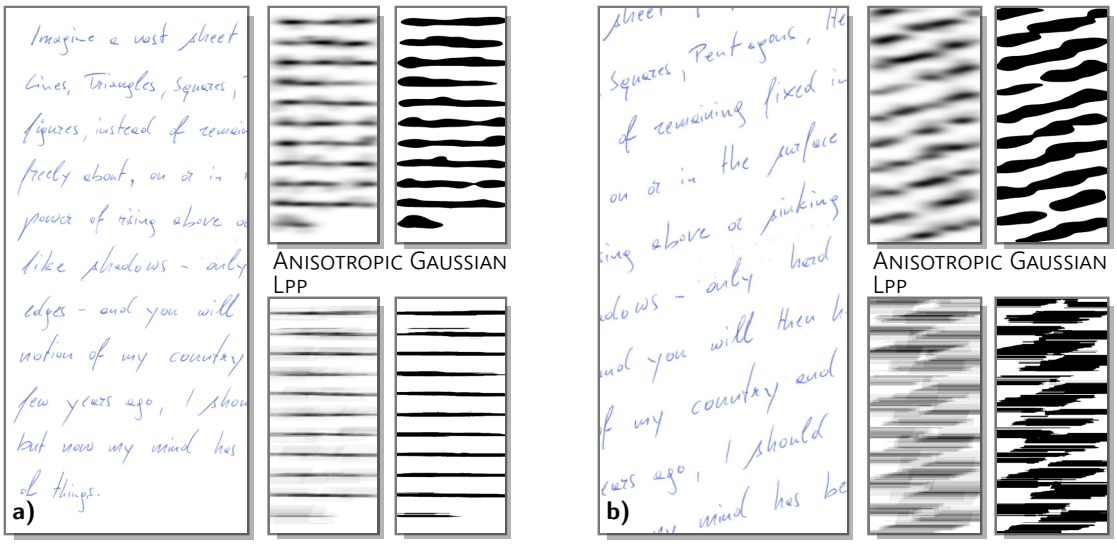


Figure 2.22: Anisotropic Gaussian filtering (upper row) and LPP of two samples.

Li et al. [YZD06; Li+08] estimate text line PDFs by means of anisotropic Gaussian filters. A level set method that forces a faster growing along the horizontal direction is

used for the final text line estimation. The level sets grow slower if small gaps between PDFs are present [YZD06]. A geometric constraint allows merging of horizontally aligned text lines while the vertical evolution between two text line PDFs is stopped if a merge is detected. In addition, they propose a post-processing step which merges text lines according to geometrical constraints such as the horizontal gap or local orientation. This method is later adopted by Du et al. [DPB09] who use morphology for the final text line segmentation.

Bar-Yosef et al. [Yos+09] propose a text line estimation of historical manuscripts. Since binarization is challenging for these kind of documents [PGN10; PGN12], they propose a binarization free text line segmentation. First the images are smeared using LPPs. Then a 1D vertical Gaussian derivative filter is applied to detect gradients in the smeared image. Text lines are segmented by detecting the zero crossings of the thus smoothed image.

Shi et al. [SSG09] use steerable filters for text line segmentation. Their steerable filter is an ellipse with a pre-defined orientation. The so-called adaptive local connectivity map gives higher responses if the current orientation fits to a text line's local skew. By these means, a smeared image is generated and text line clusters are generated with local thresholds. Connected components are assigned to those text line clusters which have the maximal overlap. A splitting rule guarantees that CCs which are connected between text lines get disconnected. This algorithm won the ICDAR 2010 and ICDAR 2009 Handwriting Segmentation Competition.

Ziaratban and Faez [ZF10; ZF11] adopt Li's approach. In order to compensate local skew variations, the filters are iteratively applied with different orientations. The local skew angle is then defined to maximize the vertical gradients of all filtered images. Elements are clustered to a text line if their distance along the local skew angle is minimal. After applying morphological operations to the thus clustered text lines, the boundaries are found which split merged CCs.

Bukhari et al. [BSB08; BSB11a] utilize an active contour approach for text line extraction of curved printed documents. Similar to previous approaches, they extract CCs in the smeared images which can be generated using LPPs, steerable filters or anisotropic Gaussians. So called baby snakes are then initialized for all CCs which are merged by means of EM. Bukhari et al. propose to use the GVF for the snake evolution of the smeared image rather than that of the input image.

2.6.2 Graph Based Techniques

Graph based techniques utilize graphs such as HMMs or MSTs for text line segmentation. In contrast to methodologies based on smearing, they do not filter the image in a pre-processing step in order to detect text line PDFs.

Koo and Cho [KC10] propose a text line segmentation for photographed books. Hence, they account for non-rigid, curved text lines and affine distortions. Having binarized the image, CCs are approximated by fitting ellipses to them. Then a local skew measure is introduced that maximizes the energy of PPs which are computed for neighboring CC ellipses. An energy minimization method segments text lines based on the

local skew and interline spacing. Additionally, elements are not assigned to a text line cluster if they violate its curvilinearity. This method is extended by Koo and Cho [KC12] for handwritten text line segmentation. For handwritten documents they introduce split moves so that merged CCs do not merge whole text lines. Again, the optimal splitting is found by minimizing the energy of PP features. This method achieved the highest performance at the ICDAR 2013 Handwriting Segmentation Contest [Sta+13].

Papavassiliou et al. [Pap+10] introduce a text line segmentation methodology that utilizes HMMs. They first split an image into vertical zones so as to minimize the effect of local skew variations. A rough estimation of text line splits is found by the smoothed PP of each vertical zone. For refining the text line splits, an HMM is constructed with two states (text, gap) and log-densities of vertical zones are the observations. Merges of CCs over two or more text lines are resolved by detecting the junctions of their skeleton. This method won the ICDAR 2007 Handwriting Segmentation Competition and is adopted by Stafylakis et al. [Sta+08] who propose to feed the HMM with PPs that are smoothed with second order Finite Impulse Response (FIR) filters. Figure 2.23 illustrates the advantage of using PPs of vertical zones rather than the whole image. The image in this figure is rotated by 15°. A Gaussian with $\sigma = 10$ is applied to the PPs similar to [Pap+10]. While text lines cannot be extracted from the PP of the whole detail, they can be directly computed from the strips.

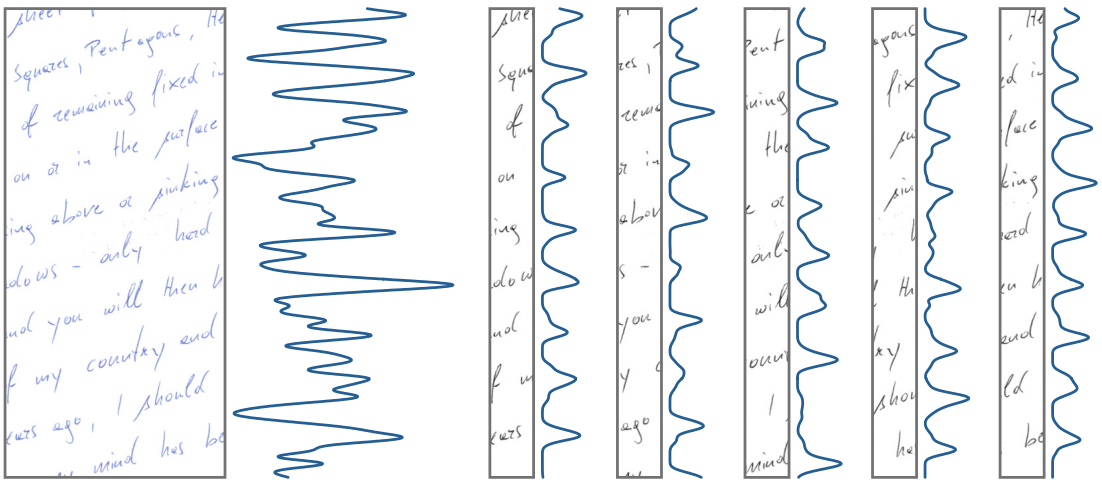


Figure 2.23: A sample page from the CVL-DB. PP of the whole detail (left) and PPs of strips (right).

Yin and Liu et al. [YL09] extract text lines in handwritten Chinese documents. They construct an MST which connects all components with their spatially connected (in terms of an area Voronoi diagram) neighbors. In contrast to previous approaches Yin and Liu use a metric that is trained. Vertically connected text lines are split by testing each edge of a cluster. The edge which minimizes the covariance matrix of the black pixels in the cluster is then deleted.

2.6.3 Other Approaches

This section summarizes techniques that neither smear text lines nor utilize graphs extracted on a document's CCs for segmentation. Both methodologies presented split coarse text line representations (e.g. words) with respect to the words' mean heights so that the segmentation process is more flexible.

Louloudies et al. [Lou+09] present a text line segmentation based on the Hough transform. They first split CCs into blocks whose width is equal to the average character height. The blocks are divided into three subsets where the first one collects characters, the second one contains characters with ascenders, descenders, or merged components and the third subset clusters small elements such as punctuations. The centers of mass of each block are transformed to the Hough domain where a voting scheme assigns the CCs to the most appropriate text line. Merges of adjacent text lines are split using a similar scheme as Papavassiliou.

Garz et al. [Gar+12; Gar+13] propose a text line extraction that is applicable for medieval manuscripts. Since binarization of noisy documents is still challenging [PGN11; PGN13], they use interest points for segmentation. The DoG interest points are clustered using Density-Based Spatial Clustering of Applications with Noise (DBscan) which – in contrast to k-Means – allows for grouping a target count estimation. A seam carving initialized with the DBscan is used to split falsely merged clusters. Text lines are then estimated by expanding minimum area rectangles similar to [DKS11]. The methodology is successfully applied to curved text lines generated synthetically in [Gar+13] where clusters are merged by means of a neighborhood graph rather than expanding minimum area rectangles. Figure 2.24 illustrates the workflow presented in [Gar+13]. DoG interest points are extracted in a). The clusters which are detected using the DBscan are split c) and then connected d) by means of a neighborhood graph. The final segmentation result e) is computed by means of the interest points' scales (colored circles).

Figure 2.24: Text line extraction using DoG interest points, courtesy by [Gar+13].

2.6.4 Evaluation of Text Line Segmentation

Similar to the Icdar Page Segmentation Competition, there is a biennial Icdar Handwriting Segmentation Contest with the objective of comparing state-of-the-art segmentation algorithms. The contest was introduced by Gatos et al. [GAS07] in 2007 and evaluates a system's text line and word segmentation performance for handwritten documents. The test data consists usually from 80 (Icdar 2007 [GAS07]) to 200 (Icdar 2009

[GSL09]) binarized document images without non-text elements (e.g. graphics, pictures). The first sample set contains two scripts – namely Latin and Greek – with writings of four different languages (English, French, German and Greek). In 2013 Bangla is added to the test data. Figure 2.25 shows sample pages from the competitions. It can be seen that locally skewed and overlapping text lines are present. Since the evaluation measure is based on a pixel level hit rate with an overlap threshold of 95%, the participating methods need to split merged text lines correctly in order to minimize the error rate. Hence, a simple CC splitting based on the global property of text lines leads to false segmentation results.

Table 2.5 details the Handwriting Segmentation Contests. Though the dataset of the ICFHR 2010 contest [GSL10] was decreased to 100 document images, its difficulty is increased if the performance of the winning method (CUBS) is compared (ICDAR 2009 99.53% and ICFHR 2010 97.63%). Even though a new script (Bangla) is introduced in the ICDAR 2013 contest [Sta+13], the performance of the CUBS method did not significantly change (97.45%). However, the state-of-the-art improved and the CUBS method was ranked 4^{th} in that contest.

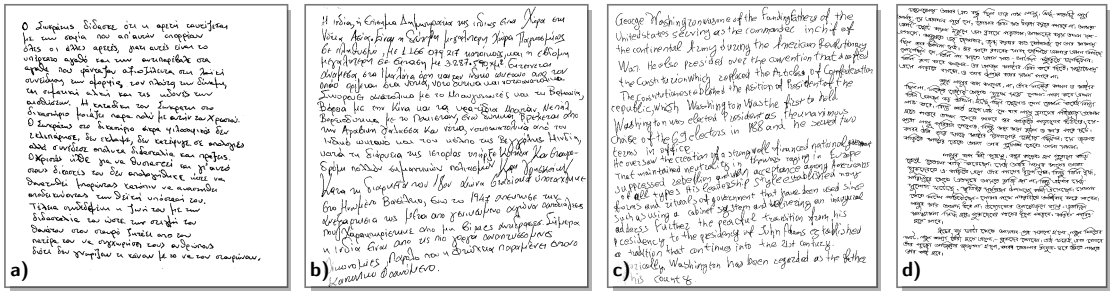


Figure 2.25: A sample from the ICDAR 2009 a) [GSL09], ICFHR 2010 b) [GSL10] and ICDAR 2013 c), d) [Sta+13] Handwriting Segmentation contest.

	#	**Languages**	#	**Winner**
ICDAR 2007	80	English, French, German, Greek	7	Papavassiliou et al. [Pap+10]
ICDAR 2009	200	English, French, German, Greek	12	Shi et al. [SSG09]
ICFHR 2010	100	English, French, German, Greek	7	Shi et al. [SSG09]
ICDAR 2013	150	English, Greek, Bangla	13	Koo and Cho [KC10; KC12]

Table 2.5: Handwriting Segmentation Contest overview. Number of test images, languages, number of participants and the winner.

2.6.5 Summary and Discussion

The state-of-the-art in document analysis focusing page segmentation and text classification was discussed in this chapter. An overview on typical features for text classification was detailed in Section 2.2. The features were discussed with respect to their invariance and distinctiveness. Text classification systems were then detailed in Section 2.4. Since there exists no benchmarking on text classification, a citation graph and an evaluation table was given so that the systems' impact can be evaluated qualitatively. In contrast to this, layout analysis systems have an international competition which allows for a quantitative evaluation. Nevertheless, it was shown that the error metric and the definition of errors is ambivalent. Depending on the application, errors might be neglected which are relevant for other applications. Hence, the competitions draw a sharp conclusion on the state-of-the-art system performance with respect to a narrow application field (e.g. good layout analysis systems might perform poorly in the Historic Page Segmentation Competition [Ant+13]). The preceding section addressed text line segmentation. It was shown that text line segmentation and layout analysis are mutually exclusive since layout analysis – if needed – implicitly segments lines. Additionally, it has been shown that the vast majority of text line segmentation approaches deal with unstructured handwritten documents. The approaches discussed are either the basics for the system proposed in this thesis or important milestones in their domain.

Considering the Page Segmentation Competitions [Ant+13; Sta+13] and recent papers such as [Gar+13; KC12], the conclusion can be drawn that modern segmentation approaches are developed beyond analyzing simple printed articles. Challenges arise either from the condition of the data (e.g. historic documents, heterogeneous supporting material, or faded-out ink), the modality of documents (e.g. heterogeneous document layouts or interfering graphic/text blocks), digitization artifacts (e.g. poor resolution, geometric distortions, or poor illumination) or because of handwriting (e.g. locally skewed text lines, overlapping text lines). Depending on the applications there are different strategies that define how the systems deal with these challenges. The next chapter outlines the proposed system and shows our strategies to cope with some of the challenges mentioned.

CHAPTER 3

Ruling Analysis

It was detailed in the introduction, that document clustering – in the context of this thesis – aims at reducing the search space for reassembling. This chapter deals with the analysis of the supporting material. The supporting material of Stasi records has on the one hand changing background color which is presented in [Die+14] and on the other hand different texture. This chapter deals with the texture analysis which can be partitioned into *void*, *lined*, and *checked* paper.

First the methodology is detailed in Section 3.1 with patch extraction, feature extraction and classification. Then a short description of the ruling line extraction is given in Section 3.1.4. Section 3.2 details the empirical evaluation. Parameter and algorithm choices (Section 3.2.1, 3.2.2) are discussed. Finally, a short summary and discussion is given.

Figure 3.1: A sample snippet of each ruling class *void* a), *lined* b) and *checked* c).

3.1 Methodology

The ruling analysis classifies snippets into *void*, *lined* and *checked*. In addition, it performs a line localization that can subsequently be used for snippet alignment if ruling

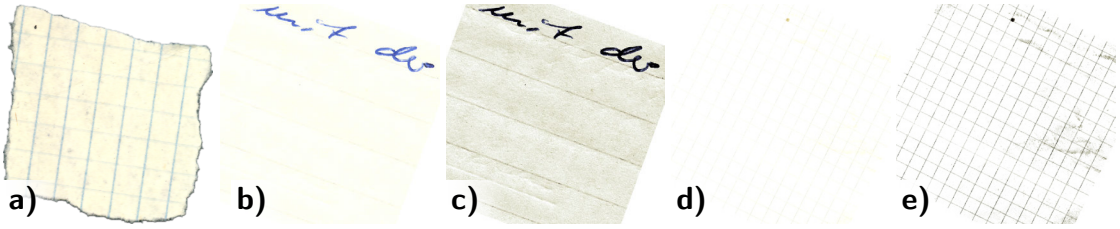


Figure 3.2: Challenging ruling examples. Different contrast between horizontal and vertical ruling lines a). Low contrast ruling lines in b) and d). After a manual histogram stretch in c) and e) the ruling lines are visible with the ancillary effect of amplified noise.

is present. Ruling has a low contrast on purpose so that humans have guides while producing a document and at the same time keeping the effect of the guides low when reading documents. In addition the contrast of ruling lines is – depending on a copier's settings – either increased or decreased when copying. Another challenge when analyzing ruling is the document's content itself. Since the content has a higher contrast (except for copies with a high contrast stretch), extraction techniques such as gradients rather detect the writing than ruling lines. A methodology to deal with these challenges is detailed subsequently.

Figure 3.1 shows an example of each ruling class. For both ruled classes (*lined*, *checked*), the ruling is clearly visible and in c) the text does not cover the entire snippet which improves the ruling classification. In contrast to this, Figure 3.2 shows challenging examples. The first snippet a) is *checked* but the horizontal ruling lines have less contrast than the vertical. The second example (b,c) is *lined* but the horizontal lines have hardly any recognizable contrast. In c) the contrast is stretched for an improved illustration. Note that the histogram stretching additionally increases the noise and renders extruded pen strokes from other pages visible. The last two images (d,e) show a similar example with *checked* paper. Here, the histogram stretching e) shows scanning artifacts that can be traced back to the low contrast.

3.1.1 Patch Extraction

Ruling lines are repeated with a fixed line frequency. For classification, a homogeneous frequency is assumed for a whole snippet. In real world scenarios local frequency changes might occur because of scanning artifacts (paper is moved while scanning) or documents imperfectly pieced together. In order to speed up the computation, a patch of maximal 512×512 px is extracted. The reason for the patch size being 512 px is discussed in Section 3.2. In order to choose a patch with maximal background in a page, an area filter is applied. The maximum of the filtered image indicates the region which contains most background pixels. Figure 3.3 shows a sample document a), its filtering result b) and the resulting image patch (blue rectangle). The area filter is visualized using pseudo color so as to increase its contrast.

A convolution using large kernels is slow if it is computed in the spatial domain. This

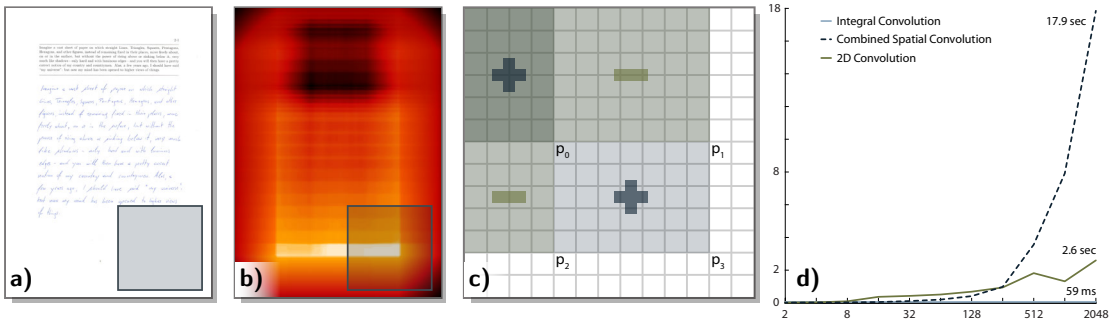


Figure 3.3: Sample from the CVL-DB a) with the resulting background patch. Corresponding area filter b). Box filter computation of an integral image c). Computation time of the three convolution techniques discussed.

can be attributed to computational complexity which is $O(N^2 \cdot M^2)$ for an $N \times N$ image convolved with an $M \times M$ kernel. Since an area filter is symmetric, two convolutions can be applied successively with a row and a column filter of length M. This reduces the computational complexity to $O(N^2 \cdot 2M)$. According to the convolution theorem [AWH05], a convolution in the spatial domain is equal to a multiplication in the DFT domain and vice versa:

$$\mathcal{F}\{I * K\} = \mathcal{F}\{I\} \cdot \mathcal{F}\{K\} \tag{3.1}$$

where $*$ denotes the convolution, I an image I, K a filter kernel and $\mathcal{F}$ the DFT. Since the DFT (without optimization) has a computational complexity of $O(N^2 \log N^2)$, convolving an image in the DFT is faster as the kernel size M increases.

Because we are dealing with an area filter – which is a box filter – the convolution speed can be further improved using integral images which were first introduced for texture mapping in computer graphics [Cro84] and later extended for general filtering tasks [VJ04]. The computational complexity of box filtering is $O(6N^2)$ additions. Hence, filtering using integral images is constant with respect to the kernel size M. In order to filter images with this method, an integral image is first computed by summing each pixel with its left and lower pixel:

$$I_i(x, y) = I(x, y) + I(x, y - 1) + I(x - 1, y) \tag{3.2}$$

where $I_i(x, y)$ is the resulting integral image of an image $I(x, y)$. In this way, each pixel value represents the area with respect to the coordinate origin. A drawback of this method is the memory usage, since the integral image needs to be allocated using 64 bit integers or floating points for 8 bit $N \times N$ images larger than $N = 2^{12} = 4,096$. Having computed the integral images, box filter operations such as the mean or area filter are computed with 4 operations (see Figure 3.3):

$$I_a(x, y) = p_3 - p_1 - p_2 + p_0 \quad where \;\; p_i = I_i(x \pm \frac{M}{2}, y \pm \frac{M}{2}) \tag{3.3}$$

Figure 3.3 d) illustrates the computation time of the three convolution techniques presented. It is evaluated in a C++ environment on an Intel i7-3520M @2.9 GHz running Windows 8.1. Each operation is carried out 10 times as a tradeoff between total testing time and stability of the results. The integral image convolution takes 59 ms for a $2,548 \times 3,510$ px image independent to the kernel size. The spatial convolution with two 1D kernels is faster than the DFT convolution up to a kernel size of $M = 256$. The 2D filtering is performed in the spatial domain until a kernel size of $M = 16$. Then, the convolution is computed using the DFT transform. The slight increase in computation time can be attributed to the cost which increases as the kernel size M grows. Though, the theoretical complexity is lower for integral images, the implementation is slower for small kernels $M < 8$ where the 1D convolution takes 32 ms.

3.1.2 Texture Feature Extraction

The subsequent feature computations are carried out on the patch previously extracted. As the contrast between lines and background is low for some images (see Figure 3.2), the gradient magnitude is computed which emphasizes edges:

$$M(x,y) = \sqrt{(I_g(x-1,y) - I_g(x+1,y))^2 + (I_g(x,y-1) - I_g(x,y+1))^2} \tag{3.4}$$

with $M(x,y)$ being the resulting magnitude image and I_g the image patch smoothed by a Gaussian with $\sigma = 3$ (see Section 3.2.2). Foreground elements are removed so that ruling lines are enhanced. Therefore, the inverted binary image (0 is foreground) is eroded with an $M \times M$ square of size $M = 10\sigma + 1$ which guarantees that the blurred edges are fully removed. A histogram stretch that maps values below the $Q_{99.5}$ Quantile linearly to [0 1] and sets values $x > Q_{99.5}$ to 1 further improves the ruling contrast. Figure 3.4 shows this enhancement for two documents a) and d). First, the gradients are computed in b) and e). Then, the foreground is removed and the histogram stretch is applied in c) and f). Especially f) shows the enhancement though the lines are noisy.

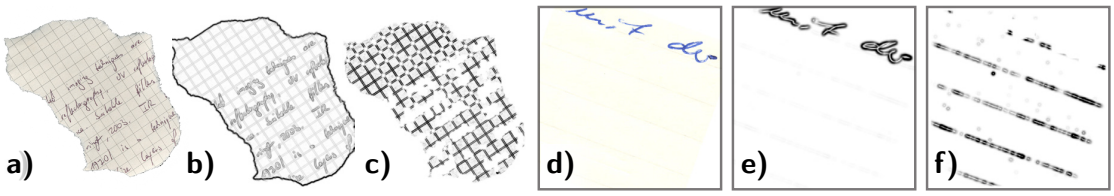


Figure 3.4: Two samples of *checked* a) and *lined* d) snippets. The gradient magnitude b), e) and the enhanced version c), f).

Lines in the enhanced patches could be extracted using e.g. the Hough transform. However, ruling – compared to other lines present in documents – has two additional properties. First, ruling lines are parallel up to a certain degree. Second, they have a fixed distance to their nearest neighbor. Since features with fixed (known) frequency and orientation can be easily extracted from the DFT, the patch is transformed to the

DFT domain. The DFT is symmetric for real-valued inputs. Hence, solely half of the DFT image is used for subsequent computations. In addition, it is shifted by $N/2$ so that low frequency are at the center of the Fourier image rather than the border. The power spectrum combining real and imaginary part is computed for the texture features:

$$\mathcal{P}(x,y) = \sqrt{Re(I_{\mathcal{F}}(x,y))^2 + Im(I_{\mathcal{F}}(x,y))^2} \tag{3.5}$$

where $I_{\mathcal{F}}(x,y)$ is the Fourier transformed image and $\mathcal{P}(x,y)$ the resulting power spectrum (see Figure 3.5 b)). The ruling orientation is unknown up to this processing stage. However, the features should be extracted rotationally invariant. That is why, the power spectrum is transformed using a polar transformation with a center located at the DFT's lowest frequencies $c = (0, N/2)$. By these means, the rows represent the changing angle θ and columns stand for the frequency. As can be seen in Figure 3.5 b), high frequencies (at the right image border) do not carry much information for these images. In addition, some areas would not be defined after the polar transform as the image is rectangular. Thus, solely 64+30 columns are considered for the feature computation. Because of the polar transform, the first columns do not contain any information needed. Hence, the first 30 columns are dropped. By these means a polar-power spectrum (see Figure 3.5 c)) is extracted from the image patch. The dotted rows in c) show the features we are interested in. The horizontal distance between the peaks is $1/f$ of the ruling frequency in the spatial domain. Since we observed *checked* paper, 90 rows (90°) are between both peaks.

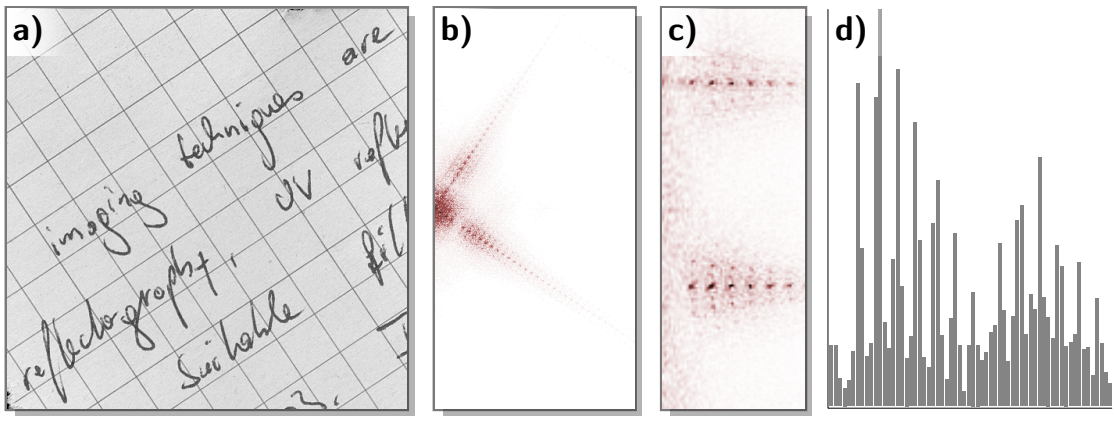


Figure 3.5: Feature extraction of a *checked* image patch. Power spectrum of the cleaned patch in b), polar transformed feature space c) and the final feature vector d).

Having extracted the polar-power spectrum, the ruling orientation is extracted using PPs. Since frequent peaks indicate the ruling, the maximum in the vertical PP is the ruling's main orientation. Skew issues of the PP discussed in Chapter 2 do not apply for the polar-power spectrum, as the features are – by definition – perpendicular to the ordinate. The row at the global maximum $max(p)$ and the corresponding perpendicular row $mod(max(p) + 90, 180)$ are extracted for the final feature vector. Concatenating

both rows results in a 128 dimensional feature vector which is normalized to [0 1] for robustness with respect to luminance changes. Instead of solely taking one row at each location, five rows are accumulated which improves the feature's profile if the ruling lines have a non-integer angle (e.g. 12.5°). Figure 3.5 d) shows a concatenated feature vector. This vector has random peaks if *void* paper is regarded, solely the first 64 bins have recurring peaks for *lined* paper and all 128 bins have peaks for *checked* paper.

3.1.3 Classification

The features are classified using a SVM with an RBF kernel [Vap82; CV95]. The classifier and kernel selection is based on tests with empirical data which are presented in Section 3.2. The training set is comparatively small with 54 training images (18 per class). The training samples were randomly chosen from an annotated data set which consists in total of 436 document snippets. The advantage of a small training set is on the one hand that training is fast[1] and on the other hand a low manual effort which includes ground truth tagging.

Since the SVM is – by definition – a binary classifier, the multiclass problem is reduced to three binary class decisions using a *one-versus-all* scheme. Therefore, three kernels are used where each corresponds to a class label *void*, *lined*, or *checked*. Hence, the classifiers are trained with all 54 sample features where the labels of the currently observed class are set to 1 and all others are set to -1. Having trained the SVMs, the prediction value for a newly observed feature vector is computed for each kernel. A negative prediction value indicates that the feature does not belong to the class of the respective kernel while features with a positive value belong to the respective class. In addition to the class label, the prediction value is the normalized distance between the observed feature vector and the hyperplane. Thus, features with low values are close to the hyperplane and therefore more likely to be wrong. The final class decision is the maximum of all three prediction values. Note that the class decision is made even if all kernels classify a feature as not belonging to their class. The 3×1 prediction value histogram is further used during re-assembling where a decision is made if the ruling feature is disregarded (depending on the likelihood of other features such as text classification labels).

In order to properly train the SVMs, a 3-fold cross validation is carried out for each kernel. Therefore, the training set is split into three equally large data sets where two are used for training and one is used for testing. This procedure is repeated until each set was once used for testing. As addressed in Section 2.3.3 the SVM – if an RBF kernel is used – has two parameters (C, γ) which need to be optimized for new training sets. The cost $C > 0$ is the penalty of a falsely classified feature vector. Low cost values result in a low penalty. The second parameter $\gamma > 0$ controls the flexibility of the RBF kernel. Here, high values allow the hyperplane to be more *flexible*, where *flexible* means that the hyperplane better fits the training data. Figure 3.6 illustrates the hyperplane with three different parameter settings. The first graph shows the hyperplane if a low cost value $C = 1$ and a low $\gamma = 1$ are used. The hyperplane is shifted away from the class with

[1] training takes ≈ 23.1 seconds including the feature computation and cross validation

more samples as the number of samples is assumed to correspond with the a-priori class probability. In addition, falsely classified variables do not have a high penalty (three training vectors are on the wrong side of the hyperplane. In the second illustration b), the cost is raised to $C = 100$ and the classification boundary shifts toward the class with a higher a-priori probability. In this example solely one feature vector is on the wrong side of the hyperplane. In the last example c) a high penalty $C = 100$ and a high RBF flexibility $\gamma = 100$ are used. The hyperplane is fitted such that no input vector is classified falsely resulting in a complex classification boundary. For this scenario, the hyperplane in a) tends to under fit the data while c) over fits it.

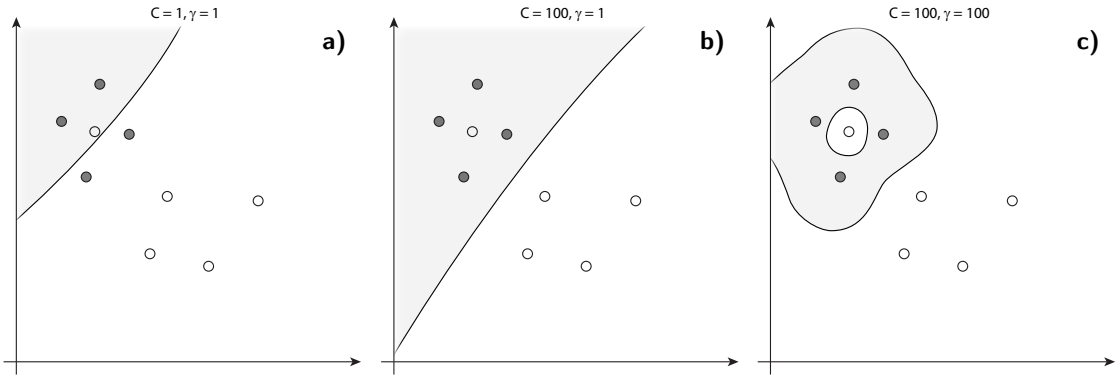


Figure 3.6: Three SVM examples with varying input parameters (C, γ).

The parameters which influence the behavior of a SVM's hyperplane were discussed previously. They are tuned for specific data using a parameter grid where each parameter is varied. The bounds are chosen to be $\exp(-5) \leq C \leq \exp(14.4)$ and $\exp(-14) \leq \gamma \leq \exp(1)$. The SVM is trained for each parameter tuple in the grid and evaluated using the 3-fold cross validation. Finally, the parameters which maximize the cross validation are chosen for training. If more tuples have the same performance, the maximum is chosen such that the cost C and γ are minimal. Figure 3.7 shows the cross validation for each kernel. The gray dot indicates the final parameter tuple that maximizes the cross validation while minimizing the parameter values. Note that the a-priori class probability is the same for all classes since the number of samples is chosen to be equally distributed.

3.1.4 Ruling Line Estimation

After classifying the document image into *void*, *lined*, and *checked*, lines are extracted for images classified as *lined* or *checked*. The ruling lines are enhanced and writing is removed as described in Section 3.1.1. Then, the image is rotated according to the ruling angle which is detected in the polar-power spectrum. A vertical PP (for documents classified as *checked* a vertical and horizontal PP) is used in order to determine the

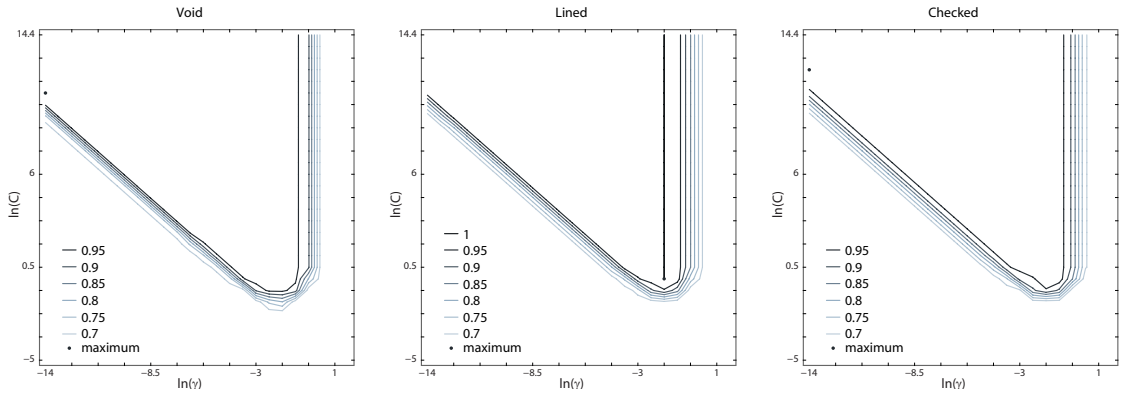


Figure 3.7: Cross validation on the training set having 18 samples per class.

accurate position of ruling lines. Local maxima p in the PP which have a stronger peak than 0.2 of the global maximum ($p \geq 0.2 \max(\text{PP})$) are used as initial guess.

A line interpolation replaces two lines with their mean if their distance is smaller than 5 px. It was previously mentioned, that ruling has a fixed frequency. Hence, the ruling frequency f_r is estimated by the $Q_{0.75}$ quartile of the distance between each line and its closest neighbor. Lines whose distance between the preceding or succeeding neighbor is not in the range of ± 10 px with respected to the frequency f_r estimated are removed. Finally, missing lines are added with respect to the frequency. Figure 3.8 shows the thus found ruling lines of two sample snippets. Green lines are those detected in the PP while blue lines illustrate *virtual* lines that are estimated based on the ruling frequency detected.

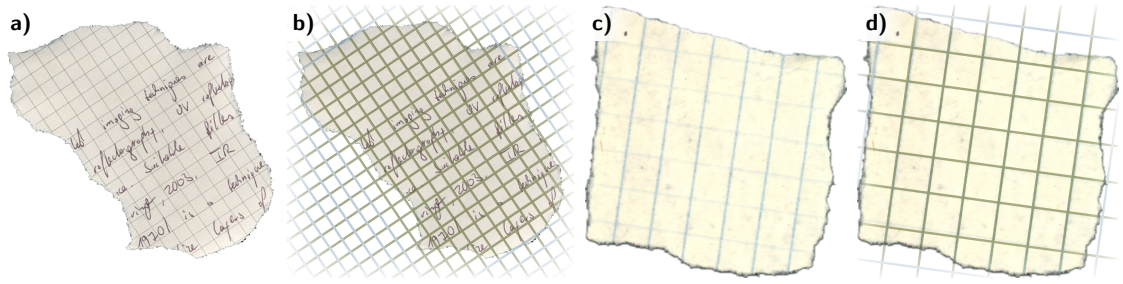


Figure 3.8: Ruling estimation of two sample snippets (a, c). Green lines are found in the PP while the location of blue lines is estimated using the ruling frequency.

The skew estimation in the polar transformed power spectrum is accurate enough for the initial line localization. However, slight deviations ($\pm 0.3°$) result in a degraded line removal. In order to overcome this, an image patch is extracted in the de-skewed background image. The height is set to 40 px while the width is the width of the currently observed line. All pixel whose vertical derivations are larger than 0.2 and horizontal deviation is below 0.2 are marked as line candidates. A robust line fitting

using the Welsch distance [WK77] is performed on these pixel. If less than 40 line candidates are detected the line gets rejected. In addition, lines whose angle difference is larger than 0.8° with respect to the global line angle, are replaced with the initial guess.

Cleaning lines is performed by a logical AND operation of the lines found and the binary image. Overlapping text elements are then removed by means of Local Projection Profiles (LPP) which are perpendicular to the lines detected. An empirical evaluation of the LPP size showed that a kernel size of 9 px has the best results.

3.2 Evaluation

The ruling estimation is evaluated on two data sets one of which has a total of 434 (SET A) and the other 511 (SET B) images. Both data sets are real world data from fragmented Stasi files scanned at 300 *dpi*. The data is particularly challenging because of its great variety. Thus, it comprises snippets with varying area, background, and background clutter. The paper fragments have a mean area of $42.4cm^2$ with a standard deviation of $\pm 37.1cm^2$ where an unsevered DIN A4 page has $623.7cm^2$. The first data set SET A has a class distribution that is more equal than that of SET B (see Table 3.1). SET B in contrast to SET A contains structured elements such as tables whose lines have a recurring distance.

	void	lined	checked	total
SET A (Train)	18 (33.3%)	18 (33.3%)	18 (33.3%)	54
SET A (Test)	143 (37.6%)	126 (33.2%)	111 (29.2%)	380
SET B (Train)	17 (32.1%)	17 (32.1%)	19 (35.8%)	53
SET B (Test)	314 (68.6%)	103 (22.5%)	41 (8.9%)	458

Table 3.1: Evaluation sets for the ruling estimation.

3.2.1 Classifier Evaluation

The SVM proposed previously for classification is compared with three frequently used classifiers, namely Naïve Bayes, LDA, and k-NN. The evaluation is carried out on SET A. It can be seen in Figure 3.9 that the Bayes and the LDA perform significantly worse compared to e.g. SVMs. This can be attributed to the curse of dimensionality. As previously discussed, the training is carried out on solely 18 samples per class while the feature dimension is set to 128. The LDA and the Bayes classifier are both *generative models*. Hence, the model parameters – whose number depends on the dimension – need to be estimated from the training samples. In contrast, the VC theory, which is the basis of SVMs, utilizes the entropy of training samples rather than their dimension [Vap06]. Thus, SVMs can find good classification boundaries even if the feature space is sparse. The same applies to k-NN which is a *predictive model* and therefore an empirical loss is

minimized rather than finding the model that generates the data. For evaluation k is set to 5.

Besides, evaluating different classifiers, three popular SVM kernels were tested on SET A. Figure 3.9 (left) shows the results. It can be seen that the polynomial kernel has the worst performance being 92.4%. In contrast to the RBF kernel which is evaluated using cross-validation, solely a polynomial kernel of 2^{nd} degree is tested. The linear SVM kernel has a lower precision by 1.05% compared to the RBF. Hence, this would be the best choice if speed is of importance as the classification can be carried out with one vector multiplication. Note that the classification is carried out once per document snippet and that the whole ruling estimation presented (including feature extraction) takes 141 ms to compute if a $2,512 \times 3,510$ px document is observed. That is why, the higher precision of the RBF kernel is chosen rather than the linear SVM.

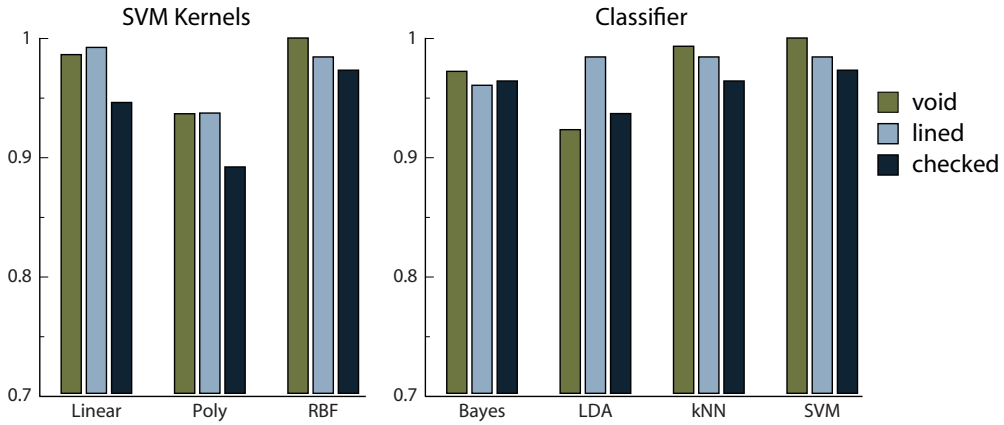


Figure 3.9: Evaluation with different SVM kernels (left) and different classifiers (right).

Furthermore the classifier's weights are evaluated to show their use for subsequent processing steps such as the reassembling. Figure 3.10 shows a plot of the precision versus classification weights in SET A. The precision is accumulated for all classes with respect to the classification weights w. For small weights, the precisions are bending because of the small number of samples. As the number of samples increases ($w \approx 0.4$), the plot gets more stable and therefore more reliable. The bins represent the underlying sample distribution. Most snippets classified have a classification weight around one which hints at a reliable decision. The plot allows for choosing thresholds on which classification results are rejected when combining different features. In more detail, if someone would need a precision of at least 0.9, a classification weight $w \geq 0.43$ has to be chosen.

3.2.2 Empirical Parameter Evaluation

The previously discussed ruling estimation has in total four parameters which influence its behavior:

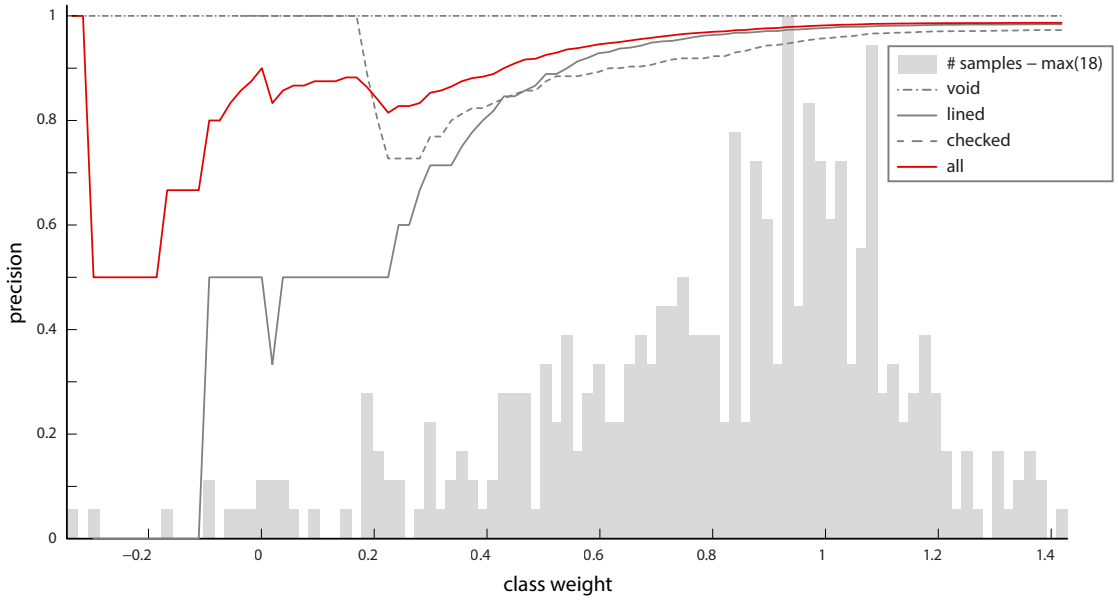


Figure 3.10: Classification weights versus precision. The precision is accumulated for each class with respect to the maximal class weight. It can be seen that the precision increases with increasing weights.

- **Sigma:** σ the filter size of the Gaussian which is applied before computing the gradient magnitude.
- **Patch Size** of the patch with most background pixels where the FFT is applied.
- **Interpolation Interval** for extracting the feature from the polar-power spectrum.
- **Feature Dimension** of the features classified.

In order to find optimal parameters for these parameters, they are evaluated on SET A. The Gaussian kernel size σ is evaluated with $1.5, 3$ and 4.5. Figure 3.11 (left) shows the evaluation. The best performance is achieved if it is set to $\sigma = 3$ which is a trade-off between removing too few noise and keeping the relevant line information. It can be additionally seen, that *lined* paper is hardly affected by σ. This can be traced back to the fact that *checked* paper has a higher frequency (lower distance between lines) and therefore peaks are merged if σ is too large. Choosing σ too low on the other hand leaves too much noise which results in a lower accuracy.

The *patch size* controls the window size which is used for observation. The maximal performance is achieved if it is set to 512 px as can be seen in Figure 3.11 (right). A too small patch (e.g. 128×128) has a bad effect on the recognition of *lined* paper because the mean line frequency is ≈ 106 px if documents are scanned at 600 *dpi*. Hence, a small patch captures solely one line which is too few for an accurate recognition. Large patch

sizes on the other hand capture more foreground information and therefore increase the risk of an erroneous binarization.

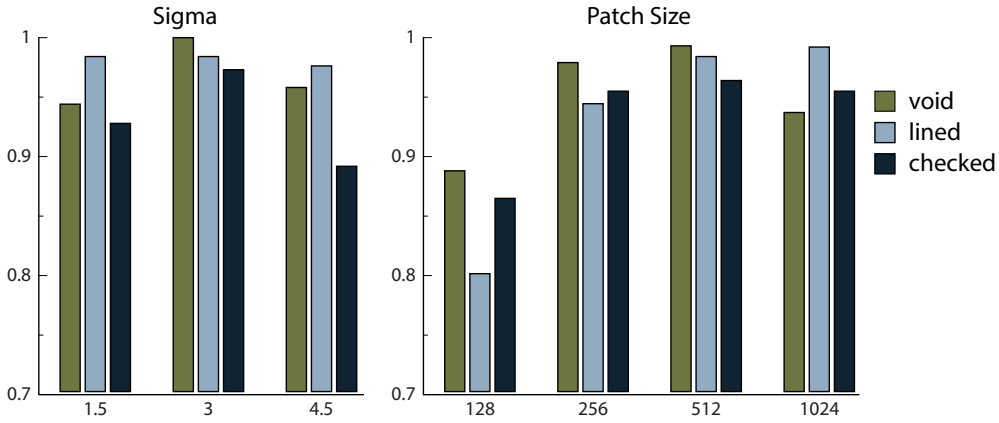


Figure 3.11: Evaluation of σ (left) and *patch size* (right).

A low *feature dimension* (see Figure 3.12 (left)) has a stronger effect on the ruling estimation (-4.7%) than a higher dimension (0.1%). Hence, the classifiers are capable of dealing with high dimensional features although the size of the training set is not increased. In addition, it can be argued from this test that a feature dimension of 64 is too low if the ruling is extracted from the FFT.

In contrast to the previously evaluated parameters, the *interpolation interval* has a comparatively low impact on the ruling estimation performance. Figure 3.12 (right) illustrates the test with varying parameters for the interpolation. An interval ≥ 1 removes noise that might be present in the line of the polar-power spectrum. Additionally, it improves the feature vector if the ruling orientation is not integer valued as it reduces interpolation artifacts from the polar transform.

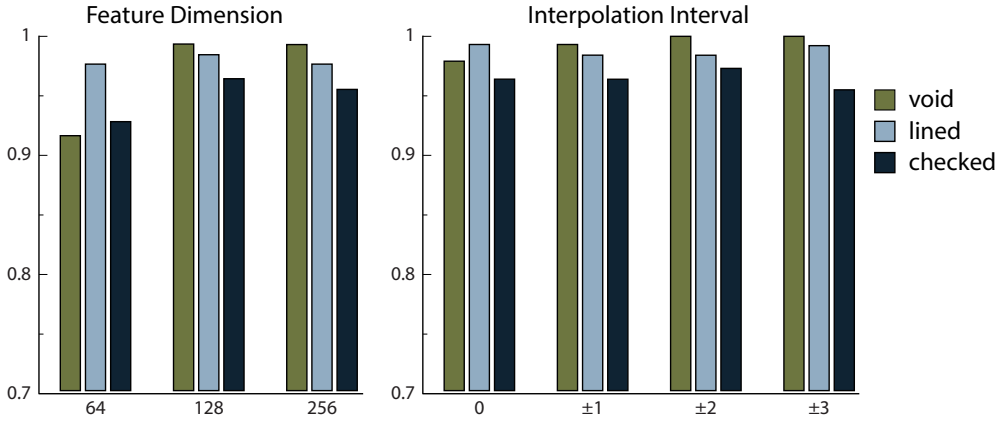


Figure 3.12: Evaluation of *feature dimension* and *interpolation interval*.

3.2.3 Dataset Evaluation

The performance of the ruling analysis presented is empirically evaluated on the two datasets previously introduced. The first dataset SET A contains clean data with no class ambiguity while the second one contains ambiguous real world examples including e.g. tables or underlined text. First, a confusion matrix is given which shows the class confusion numerically.

	a_0	a_1	$\cdots$	$\mathbf{a_i}$	$\cdots$	a_n
c_0				fp		
c_1				fp		
$\vdots$				$\vdots$		
$\mathbf{c_i}$	fn	fn	$\cdots$	tp	$\cdots$	fn
$\vdots$				$\vdots$		
c_n				fp		

Table 3.2: Confusion matrix with $n = 2$; a_i are predictions of class i, and c_i are the true class labels.

Table 3.2 shows how the confusion matrix can be interpreted. In principle, each class is plotted against each class. The rows indicate predictions while the columns represent the ground truth. Hence, diagonal elements are documents predicted as i^{th} class which are actually class i (True Positive (**tp**)). All other values are False Positive (**fp**) for the predicted class a_i and False Negative (**fn**) for the true class c_i.

	predicted			
	void	lined	checked	#
void	**1.0**	·	·	143
lined	0.016	**0.984**	·	126
checked	·	0.027	**0.973**	111
	145	127	108	380

Table 3.3: The rows of the confusion matrix show the ground truth labels of SET A, while the columns represent predicted labels (e.g. 1.6% of the *lined* paper is falsely classified as *void*).

Table 3.3 shows the evaluation results of SET A. It can be seen that using the methodology proposed with the previously evaluated parameters, all *void* classes are classified correctly. Two *lined* documents are falsely predicted as *void* and three *checked* documents are falsely classified as *lined*. Four out of the five errors can be attributed to malicious foreground extraction. The error either arises from colored ink that is not binarized or binarized lines (see Figure 3.13 a),b)) that are removed in the magnitude image. Again, the magnitude images are inverted for illustration and therefore dark areas represent high gradient magnitudes. Figure 3.13 c), d) show a scanning artifact.

In this scenario, the snippet was moved during scanning. Since the ruling lines are not parallel anymore, the proposed methodology cannot correctly classify this snippet.

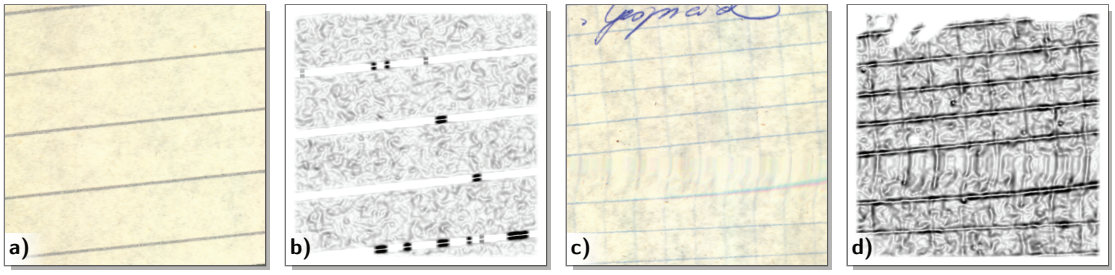


Figure 3.13: Two falsely classified examples.

	predicted			
	void	lined	checked	#
void	**0.924**	0.029	0.048	314
lined	0.116	**0.884**	·	103
checked	0.049	·	**0.951**	41
	304	100	54	458

Table 3.4: The rows of the confusion matrix show the ground truth labels of Set B, while the columns represent predicted labels.

In Table 3.4 the confusion matrix for Set B is given which is presented in [Die+14]. In this dataset *void* document snippets have a precision of 92.4%. This can be traced back to the fact that tables such as table of contents are present which have recurring horizontal and vertical lines. The comparatively low precision of *lined* documents results from empty carbon copies where lines are – similarly to the example given in Figure 3.13 a) – falsely binarized as foreground elements and therefore removed from the magnitude image.

Furthermore, F-score, precision, and recall are calculated for each class in order to allow for drawing conclusions about the nature of errors and class confusions. For these error measures, true positives tp_i, false positives fp_i, and false negatives fn_i of a given class i are defined by:

$$\begin{aligned} tp_i &\ldots \langle a_i, c_i \rangle \\ fp_i &\ldots \langle a_i, c_{j \neq i} \rangle \\ fn_i &\ldots \langle a_{j \neq i}, c_i \rangle \end{aligned}$$

where $i, j \in 0...n$ and $n = 2$ with $0 = \textit{void}$, $1 = \textit{lined}$ and $2 = \textit{checked}$. Given the true positives tp_i, false positives fp_i, and false negatives fn_i; precision p_i, recall r_i, and

F-score F_i of a class i are defined as:

$$p_i = \frac{tp_i}{tp_i + fp_i} \tag{3.6}$$

$$r_i = \frac{tp_i}{tp_i + fn_i} \tag{3.7}$$

$$F_i = \frac{tp_i}{2tp_i + fp_i + fn_i} \tag{3.8}$$

Figure 3.14 shows the precision and recall for each class on Set A and Set B respectively. The recall is low for *checked* paper in Set B being 0.722. This can be attributed to the priors in Set B where *checked* paper has a prior of solely 8.9%. This prior reflects a small subset of the real world data where printed documents are more common than handwritten. However, the true a-priori probability of all classes is unknown. That is why a second test set (Set A) is created with similar priors for all classes that reduce the bias. The recall of *checked* paper is 1 in Set A since no other class is falsely predicted as *checked*. The weighted F-score of Set A is 0.987 and 0.919 for Set B.

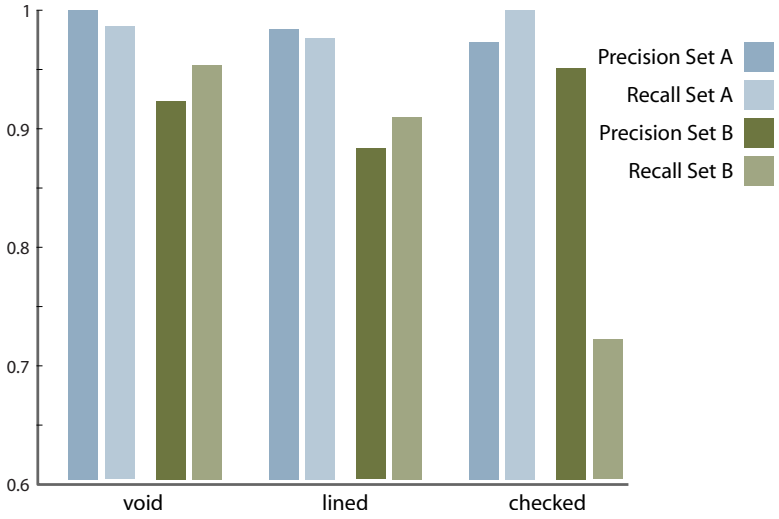


Figure 3.14: Precision recall plot of Set A and Set B.

3.2.4 Line Removal Evaluation

In addition to the classification evaluation, a line removal evaluation is performed. The evaluation is performed similar to that presented in [AKD09; LK10]. Therefore, a dataset is created by synthetically merging handwritten images from the ICDAR 2013 Handwriting Segmentation Contest [Sta+13] with four different ruling masks. The masks are extracted from scanned images and exhibit typical degradations such as broken lines. The publicly available CVL-DB ruling dataset[2] consists of 150 images written in En-

[2] http://caa.tuwien.ac.at/cvl/research/cvl-database/

glish, Greek, and Bangla. This results in a total of 600 test images (see Figure 3.15). Figure 3.16 b) shows a sample page of the dataset. Green pixel indicate **tp** (line pixel detected that are actually line pixel), red pixel illustrate **fp** (line pixel detected that are actually text pixel) and black pixel are **fn** (true line pixel that are not detected). The gray pixel are for illustration reasons, they correspond – similar to white pixel – to the **tn** class (pixel that are not line pixel and were not detected as such). Figure 3.16 b) shows that the LPP is good at removing false line detections if strokes are perpendicular to the currently observed line (e.g. upper text in the zoomed area). However, it is not capable of correctly removing false positives if text strokes are parallel to a line (e.g. lower right text). It can be seen that the methodology presented is able to detected broken and noisy ruling lines.

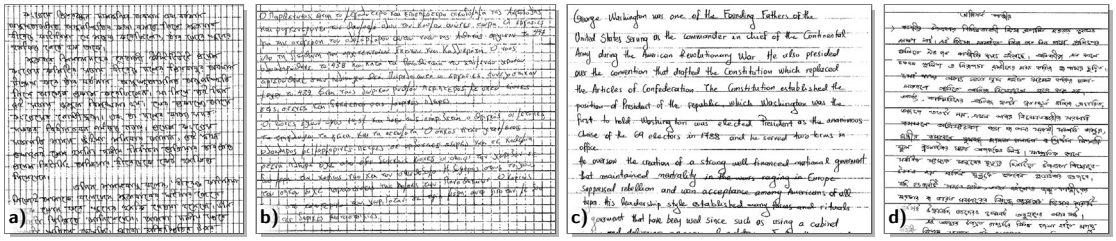


Figure 3.15: Four different samples of the synthetically generated line removal database.

The sole parameter that influences the quality of ruling line removal is the kernel size of the LPP. The Receiver Operating Characteristic (ROC) is given in Figure 3.16 a) when varying the kernel size between 0 (text restoration) and 15 with a step size of 2 px. The blue line shows the median recall versus precision when varying the kernel size. In addition the area within the $Q_{0.25}$ and $Q_{0.75}$ Quartiles is illustrated in gray. The maximal F-score of 0.93 is achieved when the kernel size is set to 9.

Since solely 9% of an image are line pixel (for checked paper), true negatives are not considered in the evaluation as this would bias the results. Hence, precision, recall, and F-score are computed for the line removal evaluation. Although the datasets are different in [AKD09; LK10] which does not allow for drawing direct comparisons, Table 3.5 gives an overview of the respective results.

	#I	**Precision**	**Recall**	**F-score**
W. Abd-Amageed et al. [AKD09]	50	0.88	0.88	0.88
D. Loprestiy et al. [LK10]	100	0.76	0.91	0.81
proposed	600	0.91	0.95	0.93

Table 3.5: Line removal comparison. **#I** is the number of images.

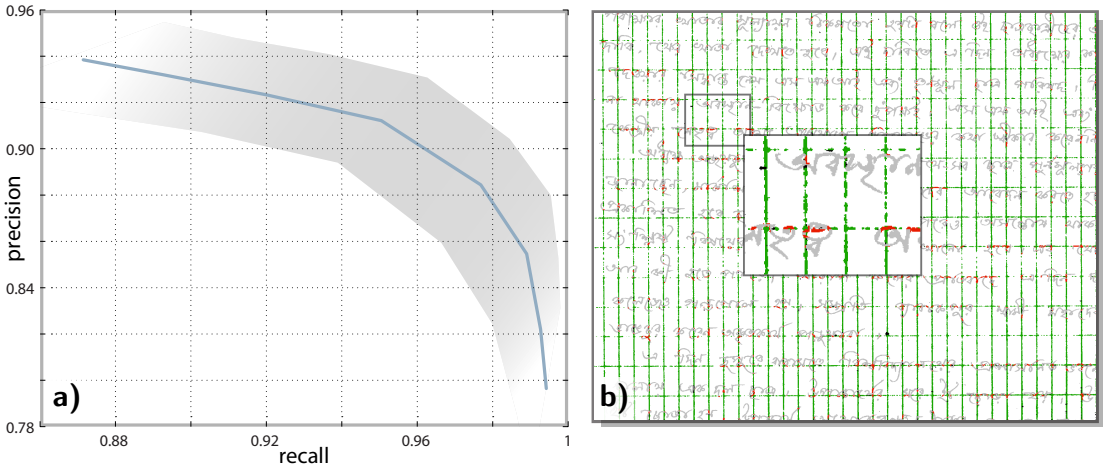


Figure 3.16: ROC curve when varying the kernel size of the LPP a). A sample page b) of the dataset presented. Green pixel are **tp**, red pixel are **fp** and black pixel are **fn**. Gray and white pixel correspond to **tn**.

3.2.5 Summary & Discussion

In this chapter, the methodology and empirical evaluation was detailed. The ruling analysis first extracts a background patch which is enhanced by means of gradient magnitudes and histogram stretching. Furthermore, foreground elements are removed for an improved frequency signature in the FFT. A polar power spectrum allows for feature extraction invariant with respect to rotation. Changing ruling frequencies are trained by means of SVMs with a one-versus-all classification scheme. The evaluation showed that the designed work flow is capable of recognizing different paper types including *void*, *lined*, and *checked* paper. Since the features are able to separate all classes cleanly, training can be carried out on a relatively small training set (18 samples per class) which reduces the human effort. The evaluation sets presented include small document snippets which show the method's robustness with respect to reduced information (e.g. short lines).

The evaluation showed, that foreground extraction (binarization) is crucial with respect to precision. Hence, foreground elements missed degrade the contrast enhancement and therefore reduce the feature's signature. In addition, binarization detects strong ruling lines if a snippet has no written content. As shown in Figure 3.13 c), the method fails if ruling lines in the patch are not parallel to each other. This is inherent with the design of the method, as snippets might be composed of several lines that are not equidistant or parallel to each other. By definition these snippets must not be classified as *lined* or *checked*. The second dataset SET B pointed out that ruling is ambiguous if formal criteria are considered. Structured elements such as tables have the same properties as ruling if they are composed of equidistant parallel lines.

CHAPTER 4

Text Classification

Text classification has a scope of application in document analysis. The methodology for text classification which is discussed in this chapter aims at annotating text in torn documents. This application and the real world data introduce a few challenges which are outlined below. The real world data are torn Stasi documents which were produced between 1950 and 1989 and are scanned at 300 *dpi*. The database contains newspaper articles, typewritten text, handwritten text, carbon copies and dot matrix printed text which can be mixed and/or overlapping within one snippet. Hence, the text classification needs to be robust with respect to noise, capable of classifying varying writing styles or fonts correctly and be able to reject false positives of the binarization. Furthermore, the document snippets range from one single word up to whole pages (if the reassembling worked out). Thus, the text classification must be capable of dealing with sparse data and missing context such as multiple text lines. In addition, there is a need for a fast computation as it is used for annotating document snippets during reassembling.

Since we deal with text, which is in general made of words that are (more or less) aligned to each other, the feature extraction is based on so called *Profile Boxes* that are located in the binary image. The advantage of profile boxes is their adoption to a word's local orientation and their robustness with respect to large ascenders and descenders in the context of cursive handwriting. A sliding window approach extracts so-called Gradient Shape Feature (Gsf) at character level. The feature extraction is implicitly robust with respect to scale changes since the profile box adopts to the word's scale. Multiple features per word further improve the classification. The Gsf are features which have a compact representation while being robust with respect to the transformations and degradations anticipated. A multi-class Svm with Rbf kernels and a one-against-one scheme is used for classification. Finally, all features of a profile box are voted to assign a class label to each box. The probability estimate histogram thus found is used in subsequent processing steps.

Section (4.1) details pre-processing steps such as dominant orientation estimation, binarization, or text line separation. Each of these modules can be disconnected if

a database does not suffer the respective degradations (e.g. for the evaluation on the IAM-DB the orientation estimation is not performed since the pages are rotated correctly). However, for the evaluation on the Stasi database all pre-processing steps are needed. Therefore errors in the pre-processing modules are reflected in the evaluation. Section 4.2 discusses the localization of text elements by means of the binary image. In addition, the computation of profile boxes is given in that section. Then, the methodology of GSF is described in Section 4.3. The training and classification of GSF features is presented in the proximate Section 4.4. The methodology presented is evaluated on four different databases. One of which consists of modern newspapers and articles, two are handwriting databases and the third one is a real world database containing torn Stasi records. The results presented in Section 4.5 show that the methodology is capable of dealing with torn documents and compares it to state-of-the-art text classification methods.

4.1 Pre Processing

Document snippets contain noise and have an arbitrary orientation. In order to compensate such artifacts, four pre-processing modules are applied before text classification. Figure 4.1 illustrates these pre-processing steps. The image is first deskewed and binarized. The binarization allows for fast region detection by means of CCs. In the deskewed image text lines are detected which are split during text classification. In addition, lines (e.g. borders of tables) are removed using the DSCC presented by Y. Zheng et al. [Zhe+01]. The pre-processing steps are discussed subsequently.

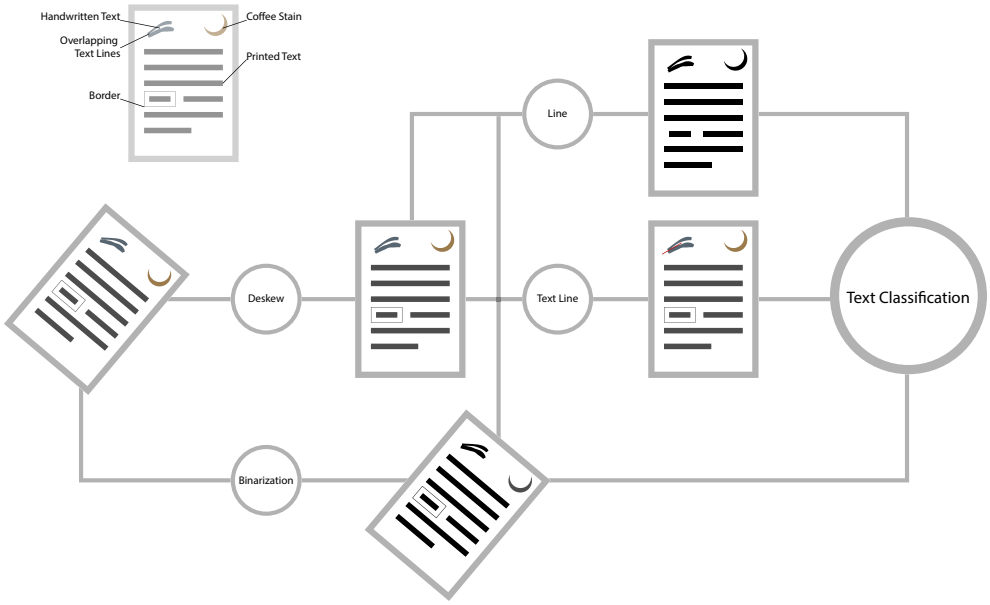


Figure 4.1: Pre-processing modules which are applied before text classification.

4.1.1 Binarization

A binarization is computed prior to text classification. The binarization partitions a document image into foreground and background regions. Ideally, all text elements would be labeled as foreground elements while all other elements would be background after binarization. However, the decision of a pixel for being labeled as either foreground or background is based on the image's gray values. Since elements such as lines, images, or stains are in the same gray value range as text, a *perfect* binarization is not possible [PGN13; PGN12].

For binarization the method proposed by Su et al. [SLT10] is extended. This method estimates the mean stroke width present in a document. Pixel are then labeled as foreground if they are within the mean stroke width of a border pixel which is detected by means of an edge map. By these means background noise such as the coffee stain in Figure 4.2 are not fully labeled as foreground which improves the results if text is in front of these artifacts.

A drawback of this method is its weakness with respect to white Gaussian noise which is present in carbon copies [KDS09]. This follows from the local contrast enhancement in the edge map which is needed for the binarization of faded out ink. To overcome this, a foreground estimation is performed which reduces false alarms in the presence of noise. Additionally, the gray scale image is combined with a saturation image which improves the detection of color text elements. Figure 4.2 shows a sample document and the resulting binary image if this methodology is applied. It is shown that the edges of the coffee stain are falsely labeled as foreground. These elements are discarded during text classification if they are labeled as background elements.

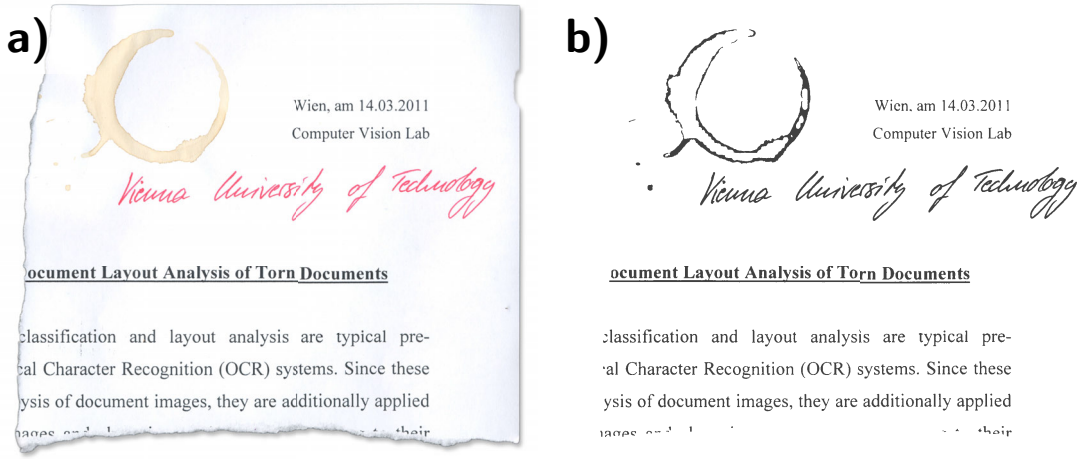


Figure 4.2: A sample snippet a) with its binary image b).

4.1.2 Dominant Orientation Estimation

The document snippets scanned have an arbitrary orientation which improves the speed of digitization (since no manual alignment needs to be performed). The text classification detailed subsequently is robust with respect to local skew changes (see Section 4.5.5). Large skew angles (e.g. if the snippet is rotated by 90°) however complicate the method's design. That is why the dominant orientation angle is estimated in advance. Modules in the subsequent computation pipeline can then use the dominant orientation to deskew the documents.

The orientation estimation which is proposed in [DKS12] combines two algorithms. The first computes the local gradient vector for each pixel (see Figure 4.3 a)) similar to the local orientation normalization proposed by D. Lowe [Low04]. The orientations thus found are accumulated to an orientation histogram with respect to the gradient magnitude. Hence, pixel located at edges are weighted more than those in homogenous regions. The global maximum of the orientation histogram is used for orientation estimation. Though this method is accurate in the presence of noise and sparse documents, it has two drawbacks. First, if handwritten text is slanted, the angle estimated represents the slant angle rather than the baseline angle. Decreasing the scale (i.e. increasing σ of the Gaussian) would reduce this effect but at the same time reduce the result's quality. The second drawback is that this method cannot differentiate between texts rotated by 90° as the characters (e.g. of printed text) possess more vertical edges than horizontal.

In order to compensate the drawbacks of the gradient orientation estimation, the FNNC proposed by Jiang et al. [JBW99] is extended. The FNNC is more robust with respect to slanted text and is capable of estimating the orientation of text rotated by 90°. If sparse documents are regarded, the FNNC's accuracy suffers.

The FNNC computes local skew lines of interest point clusters. The interest points are – in contrast to the method proposed by Jiang et al. [JBW99] – extracted using the DoG [Low99]. This allows for an orientation estimation in the gray scale image. In addition, the DoG interest point detection is fast to compute. For all interest points the k-NN with $k = 7$ are clustered. The local skew line in a cluster is defined as the connecting line of two neighbors which minimizes the distance to the interest point currently observed. Figure 4.3 b) shows an example of a k-NN cluster. The light gray line is a potential skew line which is discarded since its distance is not minimal to the interest point observed (blue point).

For a dominant orientation estimation, the angles of all local skew lines thus extracted are accumulated to an orientation histogram. Its maximal bin indicates the dominant orientation. The gradient histogram and the FNNC histogram are accumulated for the final angle estimation. To account for the method which performed more accurately on an arbitrary document, a weighting is introduced before accumulating. The weight is defined as the inverse of the histogram's area. This is motivated by the fact that the histogram's area increases if the skew detection finds multiple potential angles. This methodology achieved the 5^{th} rank in the ICDAR 2013 Document Image Skew Estimation Contest [Pap+13].

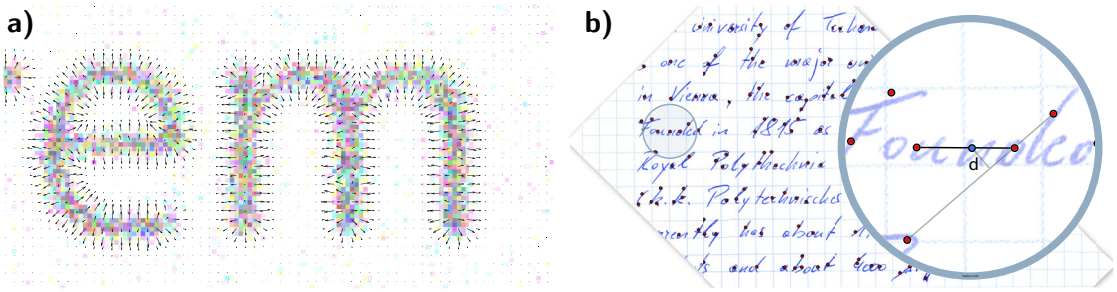


Figure 4.3: Gradient vectors of two noisy characters a). Nearest neighbors with $k = 7$, rejected local skew line having distance d and selected local skew line of the FNNC b).

4.1.3 Text Line Separation

Text lines in the binary image might be merged if ascenders overlap with descenders or the line spacing is small. Merged text lines result in poor text classification recognition rates and impair the result of text line detection or word box clustering (see Section 4.5). In order to minimize the chance of wrong text lines, a text line detection is performed. It is an extension of the approach proposed by B. Yosef et al. [Yos+09]. Since the method uses LPPs the image is deskewed with respect to the dominant angle detected. The LPP filtering merges elements which are close by and therefore reduces the effect of ascenders and descenders. An anisotropic second derivative Gaussian filter is then applied to emphasize areas between text lines:

$$G_{xx} = \left(1 - \frac{x^2}{\sigma^2}\right) \frac{-1}{\sqrt{2\pi}\sigma^3} \, e^{-\frac{x^2}{2\sigma^2}} \tag{4.1}$$

where G_{xx} is the Gaussian derivative and σ its scale. If σ is chosen too small, text lines which are close will not be split. A large σ in contrast, splits single text lines in the middle. That is why, σ is adopted for each page. Therefore, the median blob height $\overline{h}$ is computed and sigma is set to $\frac{2}{6}\overline{h}$. Choosing σ dynamically, compensates the drawbacks previously mentioned. However, text lines are still split wrongly if different line spacing is present.

Having filtered the smeared image with the Gaussian derivative kernel, local maxima are between text lines while minima are in the vertical center of text lines. Figure 4.4 b) shows a thus filtered sample image. All local maxima are set to one as initial text line estimation. It is shown in b) that homogenous regions result in spurious maxima. One could simply reject these maxima by applying a threshold. However, a rough foreground estimation similar to that presented in Section 3.1.2 allows for rejecting those maxima which are not close to the text area while keeping weak maxima in the text area. Red pixel in Figure 4.4 indicate maxima thus rejected. In addition a linearity check is performed which rejects maxima pixel that are not straight in a local context. These pixel are color coded in blue in Figure 4.4 b).

The maxima lines are not necessarily connected throughout the text area. Small gaps however result in text lines that are merged. In order to close such gaps, the maxima

lines are approximated by geometrical lines. This improves the computation speed for subsequent operations since solely the text lines need to be manipulated instead of pixel manipulations. Based on the geometrical lines, those close by with a similar angle are fused and gaps closed by these means. In addition the lines are extended which improves the splitting accuracy at the text border. Figure 4.4 c) shows the final text lines of a sample document. Note that the ascenders and descenders of handwritten text are split too. Depending on the task these can be merged after classification or simply discarded using area filters combined with the classification label.

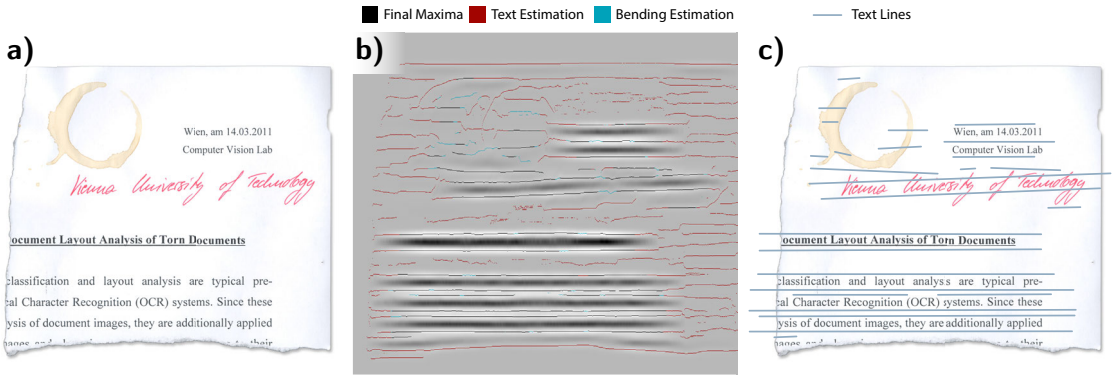


Figure 4.4: A sample snippet a), the filtered image with maxima b) and the resulting text lines c).

4.2 Text Localization

The binary image previously detailed is the basis for text localization. First small CCs with an area $a < 15\ px$ are removed. This filtering has no effect on the system's performance since these CCs are still labeled based on neighboring labels. The filtering is rather introduced to speed-up the computation. Regarding general documents with no noise, no CCs are filtered at all as the smallest assumable text element (the dot of an i) has a larger area than 15 px if the document is scanned at 300 dpi. However, if e.g. halftone prints get analyzed, the number of such small CCs can become huge which results in an increase of computational costs. Figure 4.5 illustrates the motivation of removing small blobs. The first row shows the binarization of a gray scale image. Its resolution is $3,872 \times 2,592\ px$ which is approximately the size of an DIN A4 page scanned at 300 dpi. After binarizing the image 1,596 CCs are left. This number is reduced to 495 if an area filter of 15 is applied. The second row illustrates a halftone image which are found in newspapers of the 80's that are part of the Stasi database. The computational complexity increases for this example since 48037 CCs are initially detected. The area filter reduces the CCs to 554, which is 1.15% of the initial amount, resulting in a similar computational complexity as the gray scale image. Note that filtering this images takes solely 69 ms.

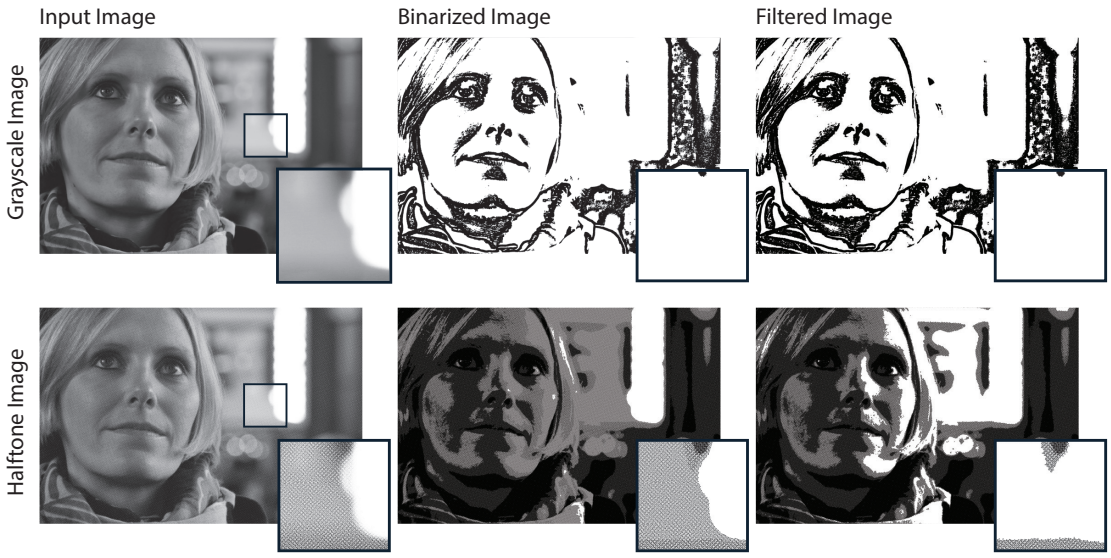


Figure 4.5: Grayscale image with its binary version and the filtered binary image (first row). Halftone image with its binary versions.

Lines such as underlines or table elements are removed from the binary image using Dscc [Zhe+01]. It is proposed in [DKS11] to fuse single characters by means of an Lpp filtering. Therefore documents with a dominant orientation $\theta > 5°$ (see Section 4.5.5) are rotated back so as to reduce the probability of wrong text line merges. Then the Lpp is applied with a kernel size of 25 which was empirically evaluated. After rotating the image back to its initial orientation a threshold of 0.1 is applied which reduces interpolation artifacts. The threshold value can be chosen arbitrarily as it is dependent to the kernel size. So increasing the threshold value and at the same time decreasing the kernel size results in the same image. As a result of the Lpp, the word blobs are elongated along the dominant orientation. Thus, an erosion with a $1 \times N$ structuring element is performed.

Recent parameter evaluations which are presented in Section 4.5.5 show that the classification performs best if no word fusion is applied. Instead, a closing is performed with a disk structing element having a radius of 3 which gave the best results on the Stasi documents. Therefore, profile boxes are computed for each cc which results in approximately one box per character for printed text and one box per word for cursive handwriting. As previously mentioned, text lines – especially in the context of handwriting – might be merged because of overlapping ascenders and descenders. Therefore, the text lines detected during pre-processing are removed from the binary image.

4.2.1 Word Boxes

For feature extraction and text line clustering a speed-up can be achieved if the CCs are further approximated since the data needed to store a CC's contour is generally more than that needed to store a rectangle. Furthermore, geometrical computations are easier – and therefore faster – if rectangles are the basic element. That is why, CCs are approximated by rectangles for all subsequent processing steps. Depending on the application, different rectangles can be used for the approximation. The three rectangle types which were proposed in [DKS11; DKS13] are subsequently discussed:

Minimum Area Rectangle is defined as the rectangle which minimizes the area of a CC. It is always fully enclosing the CC. G. Toussaint [Tou83] showed that the minimum area rectangle of a CC can be found in $O(n)$ time. Therefore, the convex hull is computed. It is known, that the polygon of the convex hull has at least one edge which is co-linear with the minimum area rectangle. Thus, it is found by scanning all pairs of orthogonal calipers and minimizing the area of these rectangles. Regarding a word, the minimum area rectangle has the advantage that it adopts to its local orientation. Ascenders and descenders of a word result in wrong local orientations (e.g. *log*). Furthermore, the rectangles become large (with sparse content) if cursive handwriting with large ascenders or descenders is present.

Oriented Bounding Box is a BB which is rotated by the documents dominant orientation. The advantage of orienting the BB is its robustness with respect to dominant orientation changes. However, if local orientation changes are present, the oriented bounding box covers an unnecessarily large area. Similar to the minimum area rectangle, it cannot compensate large ascenders in handwritten text.

Profile Box is a rectangle estimation which does not enclose the whole CC [DKS13]. It is based on the upper and lower profile line of a word which represent the word's *x-height*. Therefore, it is still compact if cursive handwriting is present. In addition, it adopts to the word's local skew. Its computation is detailed subsequently.

Figure 4.6 illustrates all three boxes for two sample images. The first image is from the ICDAR 2009 Handwriting Segmentation Contest [GSL09] while the second image is from the *Saint Gall* database [Fis+10; Fis+11]. The rectangles are illustrated in b) and d). The oriented bounding boxes are illustrated using dotted lines, dashed blue lines indicate the minimum area rectangles and filled rectangle show the profile boxes proposed. Note that the minimum area rectangle in b) differs strongly from the word's orientation. The oriented bounding box covers in both images more area than necessary and results therefore in sparse features.

For the computation of profile boxes, words are first coarsely detected by means of oriented bounding boxes. Then, the CC's upper and lower profiles are computed. Each of these profiles is observed individually and – by means of regression – a line is fit to the respective profile. Due to the ascenders and descenders, the error distribution of the

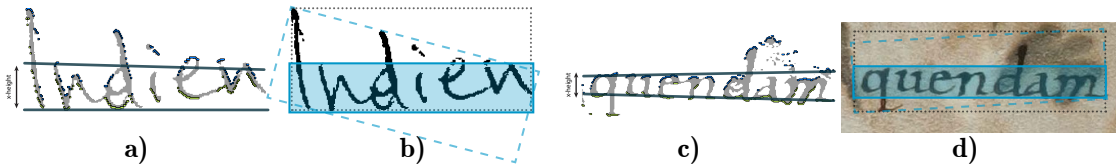


Figure 4.6: Two sample words from the ICDAR 2009 Handwriting Segmentation Contest a) and the *Saint Gall* database c). The upper (blue dots) and lower profiles (yellow dots) and the resulting profile lines. In b) and d) the resulting profile box (filled rectangle), the minimum area rectangle (dashed line) and the oriented bounding box (dotted line) are illustrated.

profiles extracted corresponds to a heavy-tailed distribution rather than a normal distribution. Hence, a robust line fitting based on the Welsch distance [WK77] is performed that compensates for large residual outliers:

$$\rho = min \sum_{i=0}^{n} \frac{C^2}{2} \left(1 - e^{-\left(\frac{r_i}{C}\right)^2}\right) \tag{4.2}$$

where $C = 2.98$ and r_i are the residuals. Even though this regression method is robust, it may fail if background clutter is present or for short words such as *"to"*. In order to handle such exceptions, the angles of the lines fitted are examined. If their difference is $\leq 5°$, a correct line fitting is assumed and the rectangle whose orientation corresponds to the mean orientation of both lines is constructed. Otherwise, the line with the minimal angle distance to the dominant orientation is considered for the profile box computation. Solely if both lines fail (their angle difference is $> 5°$), the oriented bounding box is used for further processing. The CCs of two words are illustrated in Figure 4.6 a) and c). Blue and yellow dots indicate the upper and lower profiles respectively. Lines fitted using the Welsch distance are additionally shown. Note that the noise in c) and the capital letter in a) have a low influence on the line estimation which would be worse if linear regression is applied. Figure 4.7 illustrates the profile boxes of a document snippet. In b) the profile boxes are illustrated if no LPP is applied. In contrast to this, c) shows the boxes when the LPP is set to 20. Separated boxes located at the ascenders and descenders of the handwritten text result from the text line estimation which splits CCs. The underline is not recognized since it is removed during pre-processing.

4.3 Gradient Shape Features

The text classification features are specifically designed for this task. They are adopted from Shape Context which are introduced by Belongie et al. [BMP01]. Instead of using the contours of CCs, gradient vectors are used for feature extraction. This allows for a good feature representation if noise is present that impairs the shape of contours. A famous example of using gradient vectors for local features are SIFT [Low04]. GSF have

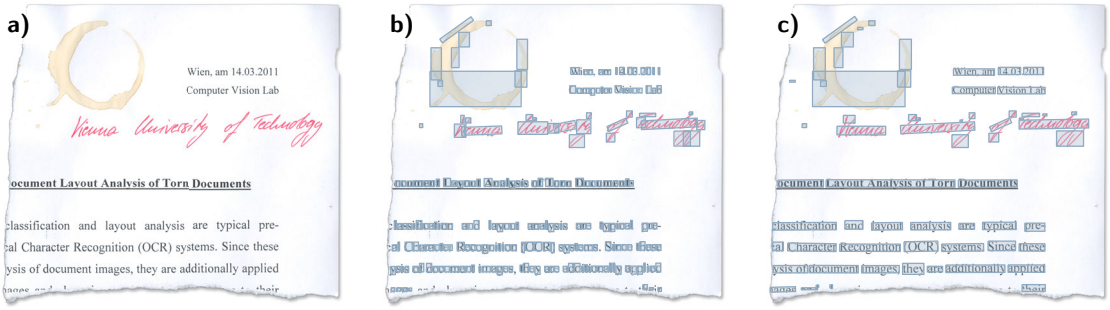


Figure 4.7: Profile boxes if the LPP is not applied b) and if it is set to 20 *px* c).

a few fundamental differences to these. First, a log-polar grid is used rather than a regular grid. Second, the orientation of gradient vectors is not accounted which allows a sparser feature representation and therefore a faster computation (the GSF has a good performance if 64 dimensions are used instead of 128). Furthermore, the feature extraction is fundamentally different to that of SIFT which uses a DOG scale-space and to Shape Context features which are extracted multiple times per CC. The feature detection – which is discussed subsequently – incorporates knowledge about the shape of text so that fewer features still lead to an acceptable classification performance. It is shown in Section 4.5 that the features proposed can compete and outperform state-of-the-art text classification methodologies.

4.3.1 Feature Detection

For feature detection, an interest point detector such as the DOG [Low04] can be used. In contrast to natural scene images, documents have the advantage that text is a structured pattern. In general, there are characters, words, and text lines. Furthermore, the orientation of characters and words correlates even in cursive handwriting documents. If this information is incorporated to the feature detection, a faster detection can be performed which localizes features on characters rather than computing a feature at every corner or junction.

For feature detection, each profile box detected is observed. A sliding window is created whose height and width correspond to the profile box's height. Note that the *height* of a profile box is the edge which is perpendicular to the dominant orientation. The sliding window is then scaled by a scale factor $s = 4$ which was empirically found (see Section 4.5.5). Increasing the sliding window especially improves the classification of small elements (e.g. dots). By choosing the sliding window's size relative to the profile box, the features are implicitly robust with respect to scale changes. Thus, the feature detection is robust with changes in the text scale without the need of explicitly computing a scale-space.

The sliding window is then moved along the rectangle's width axes with a step size of $h/2$. By these means, each pixel contributes twice to the feature computation. Tests

on the Stasi database show that the classification performance is improved if the first and last features are discarded on longer (cursive) words. This can be attributed to the fact that the beginning and the end of a word contain less structure and therefore tend to be classified as background noise. Hence, $(\lfloor 2w/h \rfloor - 2) \cdot n_\theta$ features are computed per word. Where w and h denote the width, height of the profile box respectively and n_θ corresponds to the number of dominant angles detected.

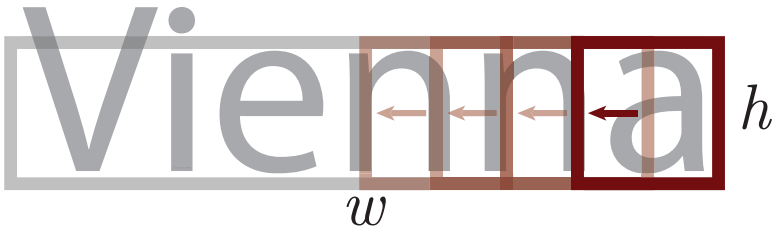


Figure 4.8: A word's profile box and its sliding window.

Orientation Normalization: The feature detection methodology proposed is by definition robust with respect to scale changes. Assuming that the profile boxes are accurately oriented, the feature computation can be normalized by the rectangle's orientation. However, neither the oriented bounding box nor the profile box guarantee an accurate orientation estimation. Since the feature computation is based on gradient vectors, discussed subsequently, the feature orientation normalization is based on these which reduces the computational effort.

The normalization is performed similar to that proposed by D. Lowe [Low04] for the orientation normalization of SIFT features. Hence, all gradient vectors which are covered by the profile box are collected. They are then accumulated to an orientation histogram having 180 bins with respect to their angle. Each vector is further weighted by means of its gradient magnitude which emphasizes gradient vectors located at edges. The orientation used for feature normalization is then determined by the histogram's maximal bin. By these means the word's slant is extracted and the features are normalized with respect to the slant rather than the baseline. In addition, this orientation is more robust in the context of cursive handwriting since each pixel represents a sample. D. Lowe [Low04] proposes to use multiple orientations if other bins are within 80% of the maximal bin. An empirical evaluation conducted on the Stasi database shows that the classification performance is reduced if multiple orientations are computed (see Section 4.5.5). That is why solely the dominant orientation θ_w is used for feature normalization.

4.3.2 Gradient Vector

In order to minimize the signal's degradation caused by background clutter or noise, a weight image is introduced. The weight image is based on the binary image which is dilated by a 21×21 square kernel k_s (see Section 4.5.5) so as to guarantee that poorly segmented strokes are still taken into account for feature computation. Subsequently, the

dilated binary image is blurred using a Gaussian kernel with $\sigma = \sqrt{5}$ (see Section 4.5) so that the weights decrease smoothly at the words' outlines. The images of all successive processing steps are multiplied by the weight image which minimizes alterations caused by noise. Figure 4.9 shows the gradient magnitude of a word from a carbon copy with noisy background. The kernel size of the structuring element k_s is varied to show its impact. If it is chosen too low, gradients can be discarded due to binarization errors. However, no weighting ($k_s = \infty$) leaves the noise at the border of the word.

Figure 4.9: A word's gradient magnitudes with varying kernel size k_s of the weight image.

Edges are important features for humans when recognizing objects [Low04]. Compared to binary features which are computed on segmented blobs, they do not suffer from alterations caused by high frequency noise. Gradients are additionally robust with respect to changes in the image's dynamic range. Thus, the features as well as the dominant word orientation are based on gradient vectors.

The gradient vectors are not computed with sophisticated edge filters such as the Laplacian or Sobel kernel. In fact they are calculated using pixel differences since this reduces the computational cost while the accuracy is not affected. Hence, the first derivatives are computed by means of a 1D Prewitt kernel:

$$d_x(x, y) = i_g(x - 1, y) - i_g(x + 1, y)$$
$$d_y(x, y) = i_g(x, y - 1) - i_g(x, y + 1)$$

where $i_g(x, y)$ represents the input image smoothed with a Gaussian. According to an empirical evaluation discussed in Section 4.5.5 it is best to choose $\sigma = 1.0$ for the Gaussian kernel. The resulting gradient images $d_x(x, y)$ and $d_y(x, y)$ are the basis for the computation of the gradient vectors $\mathbf{d}$:

$$m(x, y) = \sqrt{d_x(x, y)^2 + d_y(x, y)^2} \tag{4.3}$$

$$\theta(x, y) = \tan^{-1} \frac{d_y(x, y)}{d_x(x, y)} \tag{4.4}$$

with the gradient magnitude $m(x, y)$ and the gradient orientation $\theta(x, y)$.

4.3.3 Feature Extraction

The features proposed for text classification are based on Shape Context features [BMP01]. However, they tolerate failures of previous processing steps since the feature extraction

itself is not based on the binary image. As proposed by K. Mikolajczyk et al. [MS05], the features are robust against all anticipated transformations including changes of the word's scale, rotation, and illumination (contrast). They are not robust with respect to affine transformations which improves their discriminability.

The weighted gradient magnitude $m(x, y)$ image is the basis for the feature computation. In order to compute a GSF, solely pixel within the current character window are regarded. First, the pixel coordinates are computed relative to the center $\mathbf{c}$ of the character window which is the point of origin in the log-polar coordinate system. Then, the log-polar vector $\mathbf{p} = (r, \theta)$ is computed by:

$$r = \log \sqrt{x^2 + y^2} \tag{4.5}$$

$$\theta = \tan^{-1} \frac{y}{x} \tag{4.6}$$

with x, y being the relative coordinates of the current pixel. Figure 4.10 a) illustrates the character window and log-polar coordinates of a relative pixel vector $\mathbf{p}$. The word's dominant orientation θ_w is subtracted from the angular coordinates in order to achieve robustness with respect to rotation, resulting in $\mathbf{p} = (r, \theta - \theta_w)$.

In order to accumulate the gradient magnitudes $m(x, y)$ into the log-polar histogram, their coordinates are normalized with respect to the feature's dimension (e.g. 8×8). Interpolating the magnitudes improves the feature's robustness against noise and small variations of their location. Therefore, a linear interpolation is applied to the angular coordinates θ. Due to the log transformation of the radius a linear weighting has to be performed in the base coordinate system. Therefore a look-up table which transforms the radial bins to the base coordinate system is pre-computed by:

$$w_i = e^{\frac{i(\max r - \min r)}{N+1} + \min r} \tag{4.7}$$

where w_i denotes the *i-th* bin transformed to the base coordinate system. N is the number of bins and r represents the logarithmic radius. Having pre-computed this look-up table, the log-polar coordinates can be easily transformed to the base coordinate system, where linear weights for the interpolation are established.

Bins at the feature's border have a lower magnitude (50%) caused by the interpolation which results in a lower quality of the descriptor. In order to avoid this, a virtual bin is added to the θ axis. Thus, coordinates which are accumulated to the virtual bin, are then interpolated in the first and the *n-th* bin. For the radial dimension, 2 virtual bins are created. Pixel which should be interpolated to these virtual bins are neglected.

The distribution of the area of bins with increasing radius is not linear which leads to an inhomogeneously distributed gradient histogram. Bins near to the center have a lower gradient than those at the border. Thus, the feature's rows (see Figure 4.10 c) need to be normalized according to their area. The normalization is based on the weights w_i computed previously (see Equation 4.7), since these weights correspond to the radius of each bin in the base coordinate system. Thus, the row normalization is computed by:

$$w_{ri} = (w_{i+1}^2 - w_i^2)\pi \tag{4.8}$$

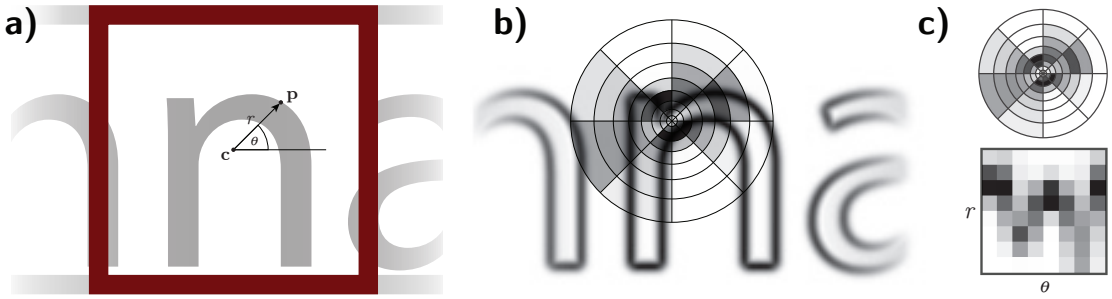


Figure 4.10: Log-polar coordinates of a pixel a), the log-polar grid on an inverted gradient magnitude image b) and the resulting feature vector c).

where w_{ri} represents the weight of the *i-th* row and w_i corresponds to the radius computed in Equation 4.7.

Considering a feature with $\max(r) = 50\ px$, the normalized **ln** is then 1.42 for $r = 2$. In other words, pixel which have a distance of 2 to the center, are accumulated into the 2^{nd} row. Thus, ≈ 8 pixel are accumulated into 8 bins of an 8×8 descriptor which results in a poor discriminability. That is why, an offset, being 10 px in the proposed system, is added to the pixels' radii which guarantees that pixel near the center still have a higher weight than those located at the feature's border. But it narrows the effect, that pixel near the center are weighted too high. Figure 4.11 shows the normalized **ln** of a feature, having a radius of 50 px accumulated to 8 bins. Note that pixel with a distance $d < 5\ px$ are accumulated into the first 24 bins. Adding an offset (shifted **ln**) to the radii results in a transformation that has a slighter bend which guarantees that more pixel ($d < 10$ px) are accumulated into the first 24 bins. The figure additionally shows the progression of linear interpolation.

Finally, a feature is created which locally captures the gradient magnitude robust with respect to orientation, scale, and contrast changes. The proposed feature qualifies for text classification since it captures the stroke width, the stroke's straightness and the appearance of junctions.

4.4 Classification

Text classification is performed using SVMs with RBF kernels [Vap06]. In contrast to the classification discussed in Section 3.1.3, the one-against-one scheme is used which improves the classification on real world data (see Section 4.5). One-against-one classifiers are introduced by S. Knerr et al. [KPD92] for digit classification. They use an NN for classification, however the rule can be adopted to SVMs. Feature vectors of each class are used for training against each other class. Hence, $k(k-1)/2$ classifiers where k is the number of classes are needed. As previously discussed, we have three classes (*noise*, *printed*, *manuscript*) which results in three SVMs. When classifying, each SVM votes for a class. The final label is assigned with respect to the class which has most votes.

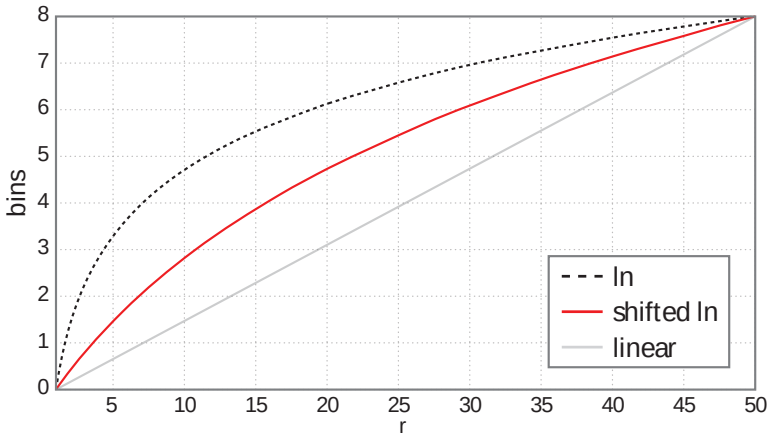


Figure 4.11: Logarithmic bin sampling. The ordinate shows the feature's bins while the abscissa shows the Cartesian coordinates in the spatial domain. Note that the first 5 pixel are accumulated to 3 bins if **ln** is applied.

W. Wu et al. [WLW04] show that multi-class classification is improved if probability estimates are used rather than the votes of the class decision. Wu et al. [WLW04] propose estimating pairwise class probabilities during training. The probability estimates are further exploited during voting of neighboring rectangles. There, overruling of neighboring rectangles is more likely if a given profile box has low probability estimates.

A cross-validation is again performed which finds the optimal cost C and γ for the training set. Figure 4.12 shows the SVM's cross-validation on the Stasi dataset. For speed-up, 2,000 features (see Section 4.5.5) randomly selected are used for the 3-fold cross-validation. It is shown that the best performance of 0.901 is obtained if $C = 4$ and $\gamma = 1$.

Depending on the database, different training schemes are exploited. On the Stasi database, the SVMs are trained on 117 document snippets. The ground truth – which is annotated semi-automatically – does not correspond to the binary image or a perfect segmentation, but rather covers text areas roughly. In order to assign a label to each profile box detected, the ground truth pixel covered are counted. A majority voting is then used to assign a label to each feature vector. Except for the minimum area threshold, all parameters are set to the same values during training and classification. The minimum area threshold is set to $t_a = 20$ (instead of $t_a = 10$ which is used for classification). This is done to discard small components whose ground truth assignment is not stable because of the small sample used for voting.

Since the binarization has a low false positive rate, too few features are trained for the noise class. Hence, training samples are not balanced which results in a biased classification. This can be compensated by increasing the weight of features labeled as noise. However, a low number of training samples results in poor generalization. That is why empty pages are used for training too. The feature extraction methodology

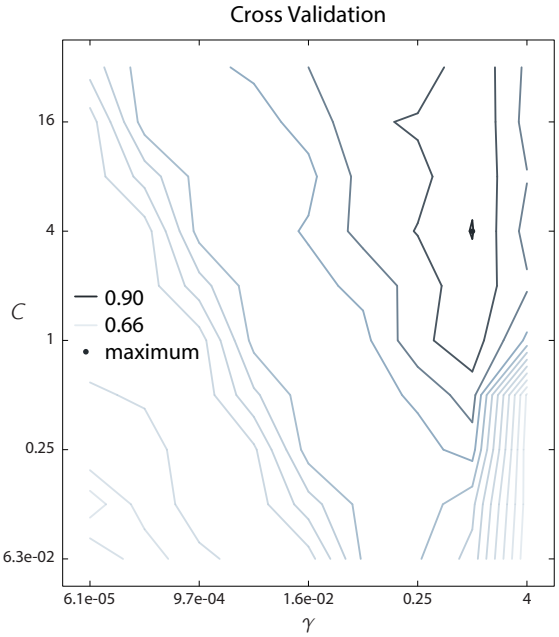


Figure 4.12: Cross Validation of the multi-class SVM. C is varied between 6.3e-02 and 32 while γ is varied between 6.1e-05 and 4.

changes if an empty page is observed during training. Therefore, features are sampled equidistantly every 7,000 pixel. The step size chosen depends on the number of empty pages. It is chosen such that the number of *noise* features is balanced with respect to the number of *manuscript* and *printed* features. In contrast to random feature extraction, the advantage of sampling with a fixed step size is on the one hand that the area is sampled homogeneously. On the other hand, the number of features extracted per empty page reflects the area of every page. Hence, the larger an empty page is, the more features are extracted. Varying scales of background elements are emulated by randomly choosing a scale between 20 *px* and 100 *px* multiplied by the feature's scale factor s. In total 21,500 features are used for training resulting in 7,860 support vectors for all three SVMs. The training on all other databases is discussed in the subsequent section.

As discussed in Section 4.2, several features are extracted for each profile box using a sliding window approach. The probability estimates of all features computed for a profile box are accumulated and divided by the total amount of features. Hence, if a feature votes for e.g. *printed*, its probability estimate for *manuscript* and *noise* are still used for assigning the class label. This guarantees that features with ambiguous probability estimates are overruled by those with strong probability estimates for one class. Finally, a probability histogram is assigned to each profile box where a bin represents the accumulated probability estimate for a specific class.

4.5 Evaluation

The text classification performance is evaluated on four different databases which are introduced below. A pixel based error metric is used to measure the performance on all databases. Depending on the objectives, additional metrics are introduced. The first database (PRImA) contains modern printed documents with complex layouts. It is on the one hand evaluated to allow for comparing state-of-the-art layout analysis systems with the text classification presented. On the other hand, it shows the capability of adopting the system to other tasks and data. In this scenario, graphical elements are trained and classified instead of handwritten text. The second database (IAM-DB) is evaluated since four state-of-the-art text classification methods were evaluated on this database. The third database CVL-DB is similar to the IAM-DB and therefore extends the insights gained from the IAM-DB. The last database is created by semi-automatically annotating real world Stasi snippets. The results show that this is the most challenging example. All pre-processing steps are needed for this database. Hence, the performance reported additionally includes errors from those modules. Parameters which are crucial for the system's performance are evaluated on the training set of the Stasi database and are further discussed at the end of this section.

PRImA-DB is created by A. Antonacopoulos et al. [Ant+09b] and is the basis for the ICDAR 2009 Page Segmentation Competition [Ant+09a]. It contains 71 document images, including newspapers with complex layouts or scientific papers. Hence, the documents contain images, charts, and tables while handwritten text is not present.

IAM-DB is created by Marti and Bunke [MB02]. It is a modern handwriting database with English texts which is designed for handwriting recognition and word spotting. In total 1,610 forms are scanned with $\approx 287,093$ handwritten and printed words. Since each form consists of machine printed text which had to be transcribed, the amount of printed words is the same as the amount of handwritten words. The results show, that the database is not challenging for text classification. The evaluation which is detailed in Section 4.5.2 is performed to give a comparison between state-of-the-art methods and the text classification system proposed.

CVL-DB is created by Kleber et al. [Kle+13]. It is – similar to the IAM-DB – a modern handwriting database. It is produced by 311 writers who wrote four English and one German text respectively. The database contains $\approx 202,138$ words. This database has no artifacts such as images or other non-text elements. Binarization can be performed using a global thresholding since the pages are scanned and the supporting material is white paper. The evaluation of the text classification presented is discussed in Section 4.5.3.

Stasi-DB consists of 426 fragmented Stasi files. It contains real world examples which the proposed system is designed for. The data is particularly challenging

because of its great variety. Thus it comprises snippets with varying area, background, and layout. The snippets have an ambiguous orientation since they cannot be aligned properly during digitization. They are written by varying type writers, scribes, and in different ink colors. Additionally, old fashioned copies with background clutter and noisy character borders are present. The paper fragments have a mean area of 42.4 cm^2 with a standard deviation of ± 37.1 cm^2 where an unsevered DIN A4 page has 623.7 cm^2. The content ranges from no content at all to a view single characters up to whole pages either handwritten, machine printed or both. Unfortunately, the original Stasi files must not be published for privacy reasons.

4.5.1 Evaluation on the PRImA-DB

Since the PRImA database is different from the mixed handwritten and printed documents which the text classification is designed for, the methodology presented is adopted for this evaluation scenario. Images within documents result in large descriptors if its height is set to the BB's height. That is why, GSF descriptors are computed at locations found by the DOG interest point detector. Then, the weight histograms of the interest points within a CC are accumulated so that a label can be assigned. Furthermore, *handwriting* is not present in this database. So instead of training a *manuscript* class, a *graphic* class is learned which includes elements such as images, charts, or illustrations.

Evaluation Metrics The PRImA database is provided with manually annotated ground truth data which is stored in XML files. Text blocks are the smallest entity which are annotated using polygons. For each text block the type is specified including text region, image region or chart region. Text regions are further specified with attributes such as reading direction, language, text color. In order to evaluate the text classification system, the attributes are converted into three different classes namely *void*, *printed*, and *graphic*. Two document samples with corresponding ground truth are illustrated in Figure 4.13. Yellow areas denote text while blue areas indicate graphical elements. Areas which are not highlighted are treated as background. In contrast to pixel accurate annotations that are used subsequently, the ground truth roughly estimates entities. For evaluation, the text line rectangles are used to label elements so that groundtruthed background elements are not recognized as false positives. CCs classified as *graphic* are labeled using their outer boundary. The evaluation metric is again computed per-pixel which allows to directly compare state-of-the-art methodologies with the proposed system.

Results on the PRImA Database All three classes *void*, *printed*, and *graphic* are trained on eight PRImA documents which are excluded during evaluation. Figure 4.14 illustrates the per-pixel precision, recall, and F-score if the 63 remaining images are evaluated. For printed text the F-score is 0.944 where precision and recall are similar (2.9%). In general whole text blocks are always classified correctly. Small text lines such

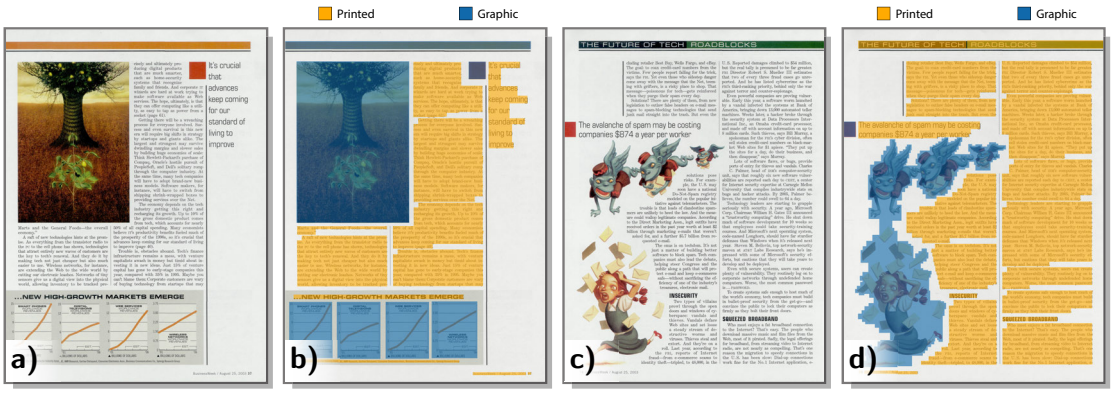


Figure 4.13: Two sample pages from the PRImA database. The yellow and blue areas indicated *printed* and *graphic* elements respectively. Note the rough approximation of the graphical element in d).

as the inverted heading in Figure 4.13 b) are responsible for false positives in the *printed* class. In contrast to this, the recall of the *graphic* class is way lower than its precision (12.5%). This indicates that the classification performance is good ($p = 0.973$) while the localization is bad for graphic elements. Localization errors can be attributed to the ground truth which does not exactly approximate a graphic's outlines (see Figure 4.13 d). Furthermore, plots or diagrams are annotated using BBs. The proposed system however labels CCs that result from binarizing the images. Thus, white areas in a plot are labeled as background which results in a high false negative rate ($r = 0.848$). Due to that, the background (*noise*) class reflects the opposite behavior with a low precision ($p = 0.917$) and a high recall ($r = 1$).

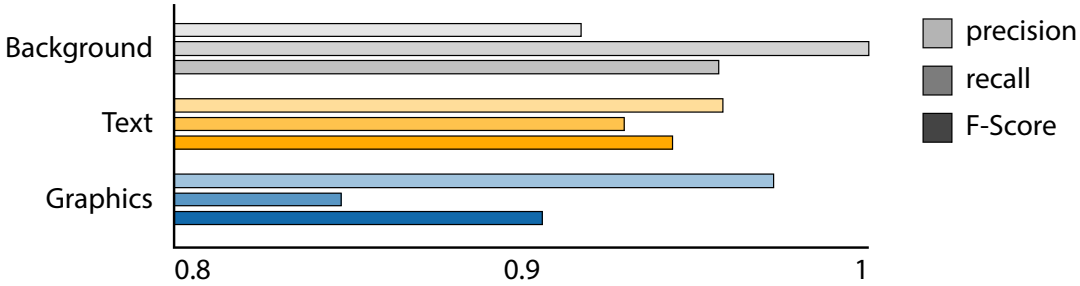


Figure 4.14: Per-pixel precision, recall, and F-score of all three classes evaluated on the PRImA database.

In contrast to other comparisons (e.g. those on the IAM-DB), the F-score $F = 0.9447$ of the proposed system can be directly compared to state-of-the-art methods which participated at the ICDAR 2009 Page Segmentation Competition since the same evaluation metrics and database are used. Figure 4.15 illustrates the *non-text, text,* and *overall* F-score. The same results are listed in Table 4.1 for a more accurate comparison. It is

shown that the Fraunhofer method [Ant+09a] has a higher precision if solely *text* elements are regarded. Since the proposed method is good at recognizing non-text elements (*background* and *graphic*) with an F-score of 0.946, the overall F-score is higher +1.3% than that of the other methods evaluated. In Figure 4.16 a sample image with the results is illustrated. Green areas are **tp** while red areas represent **fp**. Note that most **fp** in this image result from non-overlapping segmentation between the result image and the ground truth.

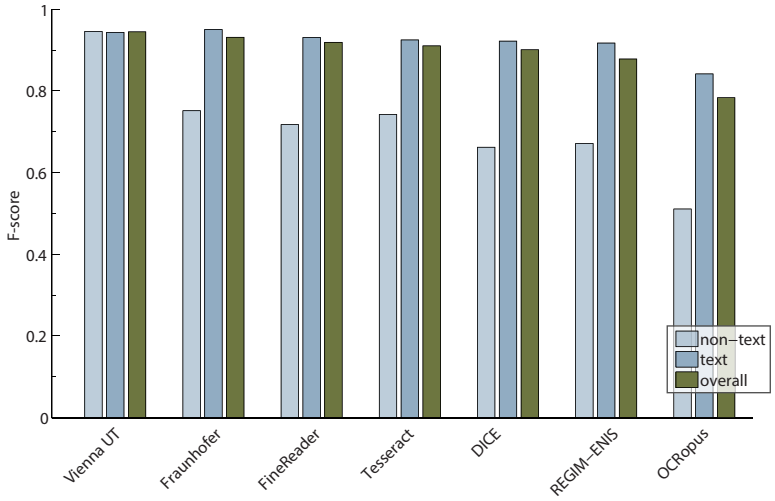


Figure 4.15: Comparison of the proposed method with all participating methods of the ICDAR 2009 Page Segmentation Competition [Ant+09a].

	Non-text	Text	Overall
Vienna UT	**94.58**	94.35	**94.47**
Fraunhofer	75.15	**95.04**	93.14
FineReader	71.75	93.09	91.90
Tesseract	74.23	92.50	91.04
DICE	66.22	92.21	90.09
REGIM-ENIS	67.13	91.73	87.82
OCRopus	51.08	84.18	78.35

Table 4.1: F-scores of the Page Segmentation Competition 2009 [Ant+09a] compared to our method (Vienna UT).

4.5.2 Evaluation on the IAM-DB

The ground truth of the IAM-DB does not directly allow for the evaluation of text classification. That is why an automated ground truth tagging is applied. Therefore, the

Figure 4.16: A sample page of the ICDAR 2009 Page Segmentation Competition with **tp** (green) and **fp** (red) annotated.

three lines present in each form (see Figure 4.17 a)) are detected using a method based on DSCCs [Zhe+01]. In total four out of 1,538 images could not be annotated automatically, because handwriting overlapped with one of the lines. The area between the first two lines is then tagged as printed text while the area between the second and the third line is tagged as handwritten. The area outside these rectangles and the area of the lines themselves are annotated as undefined since both, handwritten and printed text might be present. Then, the image is binarized using the Su et al. [SLT10] method which won the DIBCO 2009 binarization challenge. The binary image is finally combined with the tagged areas for a pixel accurate ground truth.

Figure 4.17 a) shows a form of the IAM-DB. In Figure 4.17 b) the automated ground truth is displayed. Text in the gray area (such as the form name) is not evaluated because of possible ambiguities. Figure c) illustrates the page after evaluation. In this form the "*a*" is classified falsely which results in false positives. The black pixel at the edges are false negative pixel resulting from the different binarization. As mentioned before, the ground truth is generated using the Su et al. [SLT10] binarization while the text classification is binarized with Otsu's [Ots79] method since a global binarization is reasonable for these forms.

It was mentioned in the related work (Section 2.4.4) that until now, text classification lacks a standardized dataset and evaluation method. However, since the IAM-DB is frequently used [KSA04; ZC11; Zag+12; ZL12] for evaluation, the system proposed is evaluated on this dataset. All four methods are evaluated on a different subset of the IAM-DB using different error measures. The error measures and subsets are summarized below.

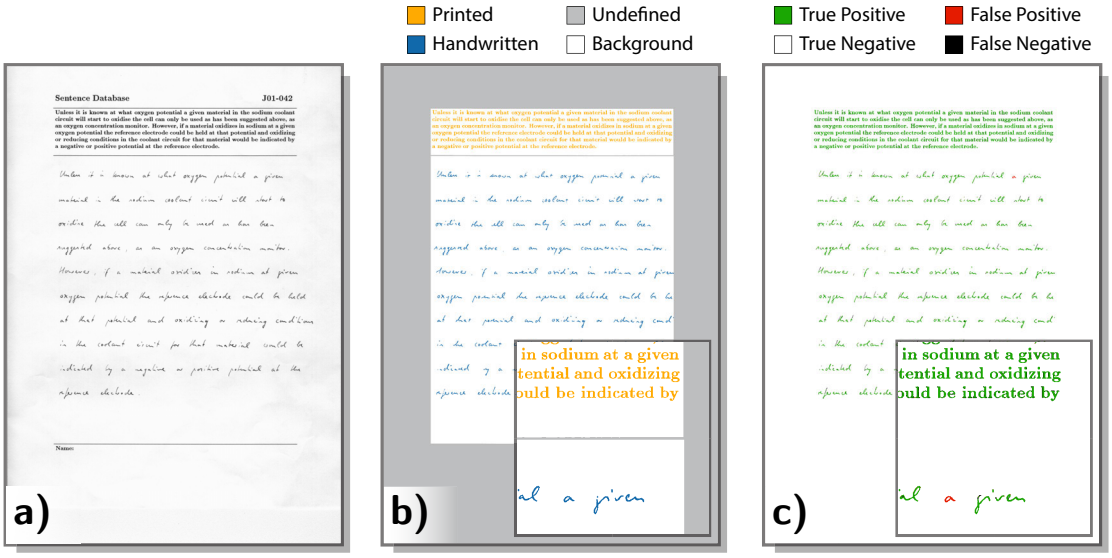


Figure 4.17: A sample page of the IAM-DB with the ground truth automatically generated b) and the result obtained by the text classification system proposed.

E. Kavallieratou et al. [KSA04] use 50 randomly selected document pages. A 10-fold cross validation is then performed for evaluation. The precision on text line basis p_t is given which is the ratio between the number of correctly classified text lines and the total number of text lines.

E. Zemouri and Y. Chibani [ZC11] use 21 images. This subset is divided into a training, a validation and a test set of equal size. Hence, seven pages are used for evaluation. Their error measure is the precision on word basis p_w which is the number of correctly classified words divided by the total amount of words.

K. Zagoris et al. [Zag+12] evaluate their method on 103 document images. Their error measure is based on the character-based F-measure [AGP10] which is further discussed subsequently.

X. Zhang and Y. Lu [ZL12] evaluate their approach on 50 randomly selected document pages. The error measure is the precision on block basis. Unfortunately, the exact measure is not detailed further.

Evaluation Metrics The IAM-DB contains text lines and words that are semi-automatically extracted. The 41,520 words and 12,662 text lines are a subset of the whole database. In order to allow for comparisons between the approaches of Kavallieratou et al. [KSA04] and Zemouri et al. [ZC11], the system proposed is evaluated on text lines and words too. The two thus introduced error measures are *precision word* p_w and

precision text line p_t and defined by:

$$p_w = \frac{\#\text{ words classified correctly}}{\#\text{ total words}} \tag{4.9}$$

$$p_t = \frac{\#\text{ text lines classified correctly}}{\#\text{ total text lines}} \tag{4.10}$$

In this classification scenario, solely handwritten text lines are present, since the machine printed text lines are not groundtruthed in the IAM-DB.

The character-based F-measure is proposed by M. Anthimopoulos et al. [AGP10] for text detection in video images. This error measure is intended to weight different text lines based on their content rather the area. The weighting function is therefore a BB's aspect ratio normalized with its area:

$$\omega_t = \frac{w}{h(w \cdot h)} = \frac{1}{h^2} \tag{4.11}$$

where ω_t is the resulting weight, w is the BB's width and h its height of either the groundtruh's BB (recall) or the BB computed (precision). This measure weights the number of pixel in a text line with the estimated number of characters in the bounding box. A BB is labeled as true positive if the amount of pixel correctly classified within a text line is more than 80%. If the amount is below this threshold, the number of true positive pixel is divided by the total number of pixel of the text line. The intention of this measure is to weight large fonts less during evaluation. If a pixel based F-score is computed, large fonts have a large area which weights these text lines more than small text lines. However, a character estimation based on the text line's aspect ratio incorporates assumptions [AGP10] such as equidistant character spacing and fixed character width which is not valid for printed text in general. Considering handwritten text, the approximation is even more critical. Figure 4.18 shows the same word written by two different writers and with two different fonts. The right sample in the first row uses a monospaced font namely Courier New that would comply with the assumptions in [AGP10]. In this scenario the weight $\omega_t = 5.44$ is close to the real character count (6). However, these examples show why the character-based F-measure is not ideal for the scenario of handwriting evaluation. While the pixel based F-score has a standard deviation of $\sigma = 0.19$ for these example images, the character-based F-measure has $\sigma = 0.32$. Note that $\sigma = 0$ is desired in this scenario since the four BBs have the same textual content.

That is why, a pixel based F-score is used for evaluation rather than the character-based F-measure. **tp**, **tn**, **fp**, and **fn** are therefore defined by:

- **tp** true foreground pixel which have the correct class label.
- **tn** true background pixel labeled as background.
- **fp** pixel which are labeled wrongly (either as printed or handwritten).
- **fn** true foreground pixel which are labeled as background.

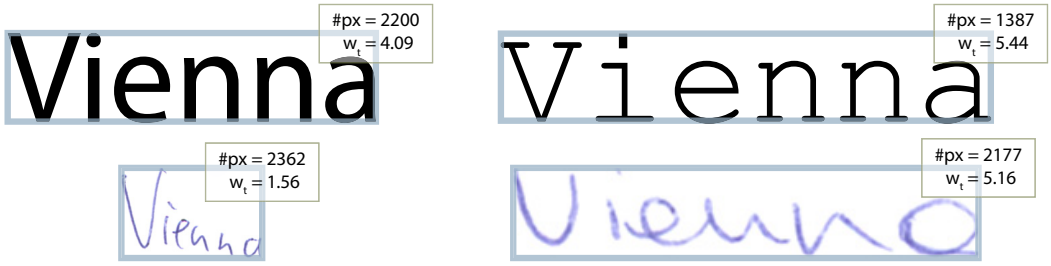


Figure 4.18: Two printed examples (upper row) and two handwritten examples (lower row) with the CC area per BB (#px) and the BB ratio ω_t proposed.

based on these definitions, precision p, recall r and F-score F are defined as in Section 3.2.

In addition to the pixel based F-score, a word based F-score is used. For this measure CCs are merged using an LPP which approximate word BBs. The performance metric is based on a MatchScore [GSL10] which considers words as **tp** if the ratio between **tp** pixel and total foreground pixel is more than a certain threshold T_α. The threshold T_α is set to 90% which is the same as in [GSL10]. Hence if more than 10% of the pixel within a word are **fp** or **fn**, a word is tagged as **fp** or **fn** respectively. This measure has the same intention as the character-based F-measure with the difference, that all examples in Figure 4.18 have the same weight. The word based measures are denoted by p_{ms} (precision), r_{ms} (recall) and F_{ms} (F-score).

Classifier Training: In contrast to tests performed on the PRImA or Stasi dataset, the classifier is trained on two classes (*printed*, *manuscript*). The noise class is not trained for the IAM-DB since the images can be correctly binarized using a global threshold. CCs with an area below 300 pixel are not trained since they represent artifacts or punctuation marks. In total, 72 images randomly selected compose the training set. The training set size of 72 images is a trade-off between speed and accuracy. In order to demonstrate that the impact of the size is negligible, a test is performed using solely one page for training. Figure 4.19 compares the precision if different training and test sets are chosen. The transparent bars illustrate the word precision, while the opaque bars show the pixel precision. The first test is the full evaluation performed using 72 training and 1,534 test images. If 35 images are selected for testing, the word precision is $p_{ms} = 0.998$. When the training set is reduced to solely one page, the precision drops to $p_{ms} = 0.963$. Though this indicates that 214 additional words out of 6,043 are classified falsely when the training set is reduced by 98.6%, it can be concluded that the system's performance is not critical with respect to the training set size.

Results on the IAM-DB: The evaluation on words semi-automatically extracted is carried out on 46,184 words. Unfortunately, this dataset contains not only words but also words canceled by the writer and punctuation marks (see Figure 4.20 (right)). The text classification's precision on all words is $p_w = 0.909$. Since punctuation marks are

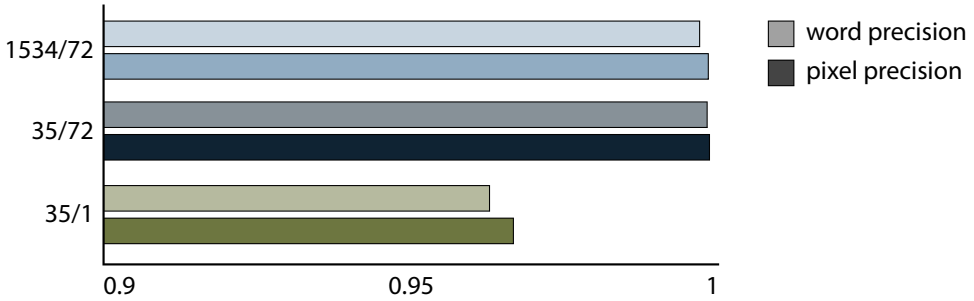


Figure 4.19: Precision with changing training and test set size. The numbers right of the plot denote the test and the train set respectively (test/train).

rejected during training, the majority is falsely recognized as printed text. In addition, the text classification is designed for larger blobs which renders a correct classification of small blobs difficult. Figure 4.20 illustrates this problem. The red curve shows the precision accumulated with increasing word size. The word size in this scenario is defined as the BB's area of a word. It is shown, that the word distribution is skewed towards small words (gray bins). With increasing word size, the system's precision increases and converges towards 0.91. In order to give more reliable results, words with an area below 1,000 pixel, are rejected during evaluation. The threshold is chosen such that the majority of punctuation marks is rejected while the smallest English word 'a' is still evaluated. If this threshold is applied, the precision is increased by 0.07 to $p_w = 0.979$. The evaluation on words semi-automatically extracted is carried out on 46,184 words. Unfortunately, this dataset contains not only words but also words canceled by the writer and punctuation marks (see Figure 4.20 (right)). The text classification's precision on all words is $p_w = 0.909$. Since punctuation marks are rejected during training, the majority is falsely recognized as printed text. In addition, the text classification is designed for larger blobs which renders a correct classification of small blobs difficult. Figure 4.20 illustrates this problem. The red curve shows the precision accumulated with increasing word size. The word size in this scenario is defined as the BB's area of a word. It is shown, that the word distribution is skewed towards small words (gray bins). With increasing word size, the system's precision increases and converges towards 0.91. In order to give more reliable results, words with an area below 1,000 pixel, are rejected during evaluation. The threshold is chosen such that the majority of punctuation marks is rejected while the smallest English word 'a' is still evaluated. If this threshold is applied, the precision is increased by 0.07 to $p_w = 0.979$. In addition, the precision is computed on 12,662 text lines which are semi-automatically extracted. For text lines, the text classification achieves a precision of $p_t = 0.996$. It is shown, that the performance is increased if more information is provided at once.

In addition to the tests performed on words or text lines extracted, the text classification's performance is measured on whole pages. Here, the ground truth automatically generated is used. A perfect segmentation is simulated by assigning the class labels to the

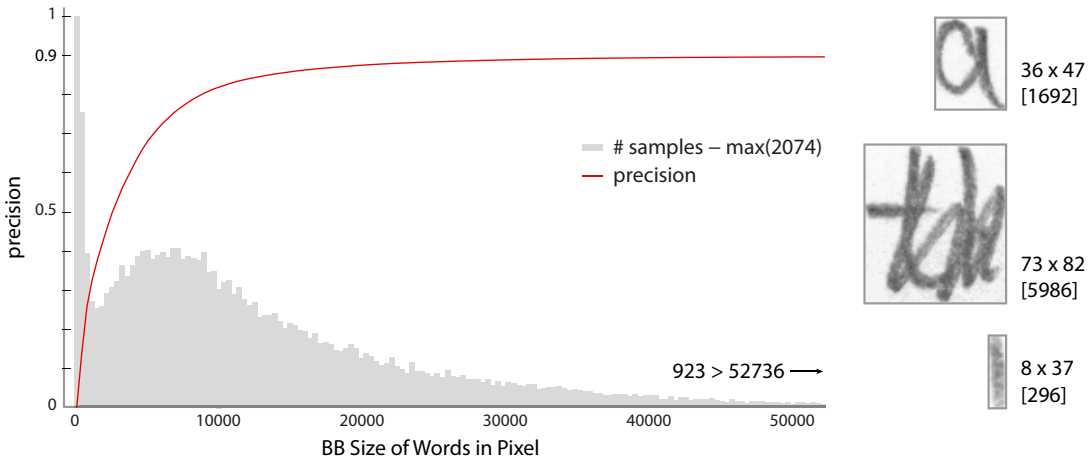


Figure 4.20: Accumulated precision (p_w) with respect to the word's BB size. The bins show the size distribution. The three sample images on the right illustrate the smallest possible word in English '*a*', a canceled word (middle) and a comma.

same binarization as the ground truth is generated with. In addition, the performance drop is computed if the forms are evaluated using Otsu's [Ots79] binarization method. If a perfect segmentation is assumed (solely the classification is performed in this scenario), the text classification achieves an F-score of $F = 0.998$. The equivalent word error is $F_{ms} = 0.996$ on 287,093 words. Figure 4.21 illustrates the results. The first two blocks show precision, recall, and F-score evaluated with a perfect segmentation. The recall is in this case always one because **fn** cannot exist due to the perfect segmentation. The two lower blocks again show the word and pixel evaluation when Otsu's [Ots79] method is used for binarization. This figure additionally shows the difference of the two performance metrics (F, F_{ms}). For the perfect segmentation (no Seg), the F_{ms} slightly decreases (0.14%) compared to the pixel based F-score. This can be attributed to the fact that no **fn** are present in this evaluation. Since the F_{ms} has way less entities than the pixel based F-score ($2.87 \cdot 10^5$ vs. $3.37 \cdot 10^8$), a single word which is falsely classified is more important than a falsely classified pixel. In addition, as shown before, short words such as an '*a*' are more likely to be classified wrong. These words have a smaller area which additionally weights them less in the pixel based F-score. If Otsu binarization is used, the F_{ms} has a higher performance ($F_{ms} = 0.968$) than the pixel based F-score ($F = 0.938$). In this scenario, border pixel are likely to be false negatives, since the Otsu binarization tends to undersegment words. Hence, the F_{ms} is more appropriate here as it emphasizes classification errors or word misses rather than little changes between the binarization methods.

The text classification is compared to state-of-the-art methods in Table 4.2. Unfortunately none of the authors evaluated their method on all IAM-DB forms. Thus, the results cannot be directly compared. For the proposed method, all evaluation measures previously discussed are given such that the performance can be compared with each of

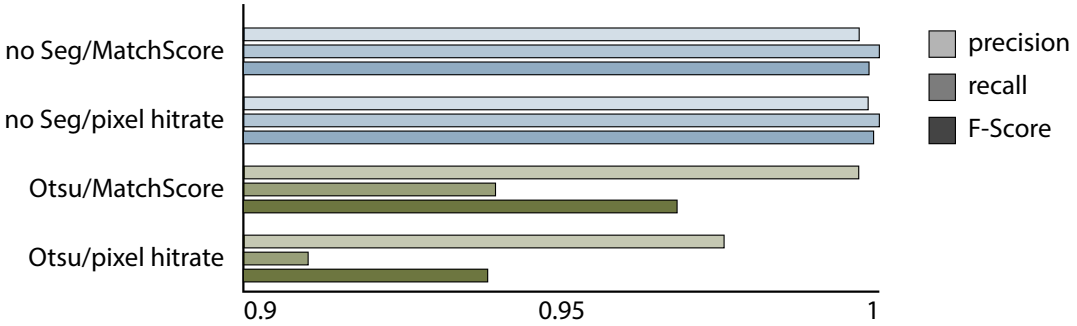


Figure 4.21: Precision, recall, and F-score of the text classification when evaluated on all 1,534 pages of the IAM-DB. The first two blocks show the performance without segmentation while the second two blocks illustrate it with imperfect segmentation.

the methods listed. It is shown that the performance is generally high ($> 97\%$) though the IAM-DB has 657 different writers. The high performance can be attributed to the fact that the database is clean in the sense that there are no artifacts, non-text elements or other degradations. Table 4.2 shows that the proposed text classification system can compete with state-of-the-art methods. Note that the method of Zemouri et al. [ZC11] has a higher performance by 0.4% indeed, but it was evaluated on solely 7 pages. It was shown before, that the performance of the proposed system increases when the dataset is decreased. In addition, they tagged their dataset manually and may therefore not suffer from noisy data such as punctuation marks or words canceled.

method	#	p_w	p_t	F_{ms}
Kavallieratou [KSA04]	50		98.2	
Zemouri [ZC11]	7	98.3		
Zagoris [Zag+12]	103			98.9
Zhang [ZL12]	50			99.9
proposed no Seg	1,534	97.9	99.6	99.8

Table 4.2: Comparison with state-of-the-art text classification methods on the IAM-DB. The F-scores are in %.

4.5.3 Evaluation on the CVL-DB

The evaluation on the CVL-DB is carried out in the same manner as that of the IAM-DB. Again, the ground truth is automatically generated by detecting the separator lines in the form. Figure 4.22 shows a sample image from the CVL-DB. It is shown in a) that the form is slightly different to that of the IAM-DB. The groundtruth b) is again generated using the binarization of Su et al. [SLT10]. Figure 4.22 c) shows the resulting text blocks and text lines. The writer ID is falsely classified as handwritten since it is not trained. In addition one quote ([...] *universe"* [...]) is not merged to any text line and

is thus recognized as isolated text line. However, such errors can be easily handled by e.g. rejecting text lines whose area is below a certain threshold.

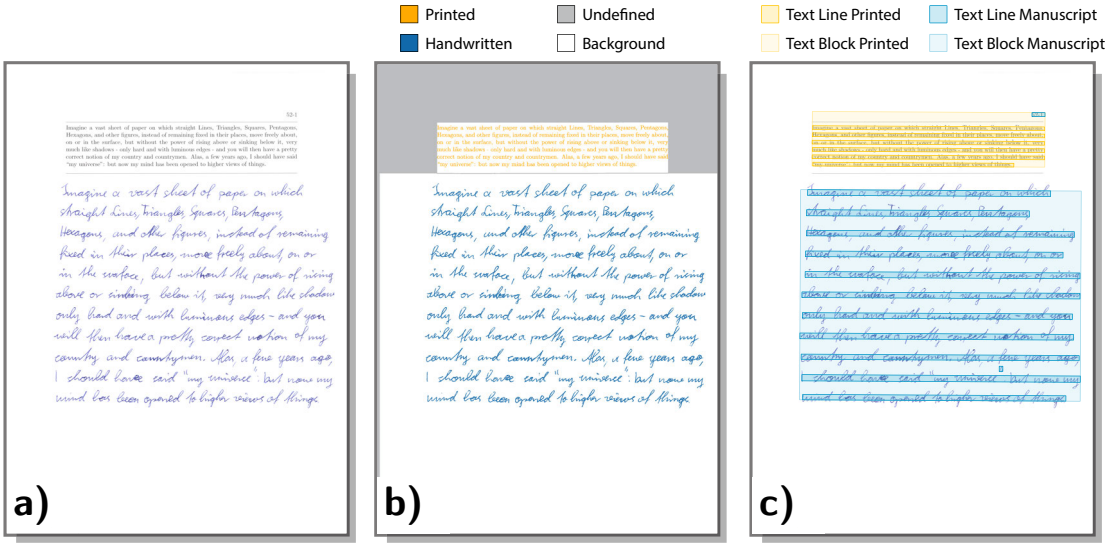


Figure 4.22: A sample page of the CVL-DB a), the ground truth automatically generated b) and text lines automatically extracted c).

The SVM is again trained on two classes (*printed*, *manuscript*). The official CVL-DB training set consisting of 189 forms is used for training. Figure 4.23 shows the results compared with those of the IAM-DB. The first two rows illustrate the F_{ms} and the per pixel F-score on the IAM-DB. The two lower rows show that the performance on the CVL-DB is improved. The improvement can be traced back to two differences between the IAM-DB and the CVL-DB evaluation. First, the training set of the CVL-DB is larger (189 images) than that of the IAM-DB (72 images) because the official training set is used. And second, the IAM-DB has different texts, while those of the CVL-DB are the same for every writer. Table 4.3 shows the results numerically. It is shown that the CVL-DB is slightly smaller. In both scenarios the p_{ms} and per pixel precision are equal up to the third decimal place. However, Figure 4.23 shows that the p_{ms} is slightly lower than the per pixel precision.

database	# pages	# words	p_{ms}	p
IAM-DB	1,534	287,093	0.998	0.998
CVL-DB	1,415	248,497	0.999	0.999

Table 4.3: Number of pages, words, precision MatchScore and per pixel precision on the IAM-DB and CVL-DB respectively.

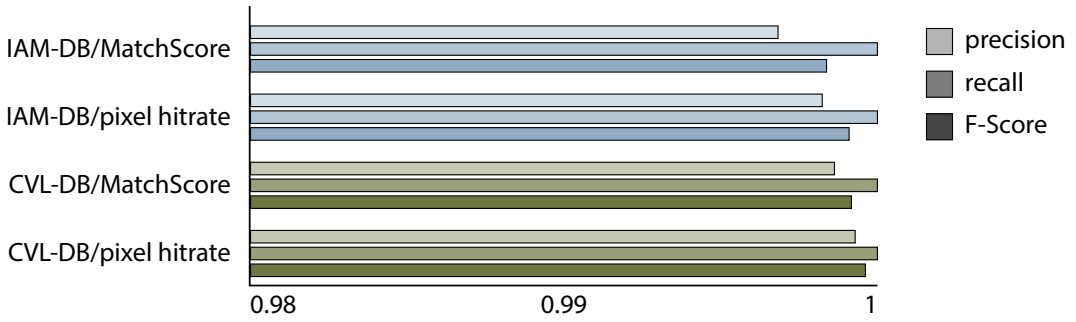


Figure 4.23: MatchScore and pixel hitrate on the IAM-DB and CVL-DB respectively.

4.5.4 Evaluation on the Stasi Database

The original Stasi database is the most challenging database evaluated. It was mentioned before, that it consists of 426 document snippets from different sources and decades. Some copies have heavy background clutter and the ground truth is not always correct because handwritten text overlaps with printed text. Additionally, non-text elements such as drawings or lines are present. For evaluation, the parameters which are discussed subsequently are used in order to achieve the best performance. Figure 4.25 shows two sample images which are not from the Stasi database but illustrate some challenges of it. A carbon copy is illustrated in a) with noisy and blurred edges along the character borders. The second sample b) illustrates typewritten text with handwritten annotations. Green rectangle represent boxes which are correctly classified while red rectangles denote false positives. Minimum area rectangles are computed rather than profile boxes. Note that they have wrong orientations for short words such as "*of*".

The ground truth is semi-automatically generated. First, a smearing is applied to CC blobs so as to minimize the human effort. Then, human operators label each thus fused CC. If background is falsely detected, it is removed. Missing foreground elements are added too using Photoshop. Hence, the ground truth is not pixel accurate and parts of ascenders/descenders might be missed.

For evaluation, the error introduced by binarization is again minimized by combining the ground truth with the automatically extracted foreground elements. Then again, the per pixel error measures including precision, recall, and F-score are computed. Since the system is capable of rejecting potentially false foreground elements, the **tn** are computed too. Note that **tn** are solely those pixel which are labeled as foreground elements but rejected during classification. All other *off* pixel are not regarded during the evaluation since they represent the vast majority (96.01%) and would therefore bias the results. The accuracy a is defined by:

$$a = \frac{tp + tn}{tp + fp + fn + tn} \tag{4.12}$$

In addition to precision and recall, the accuracy incorporates both error types (**fp**, **fn**) at once. The second error measure is based on the rectangles. Therefore a rectangle is

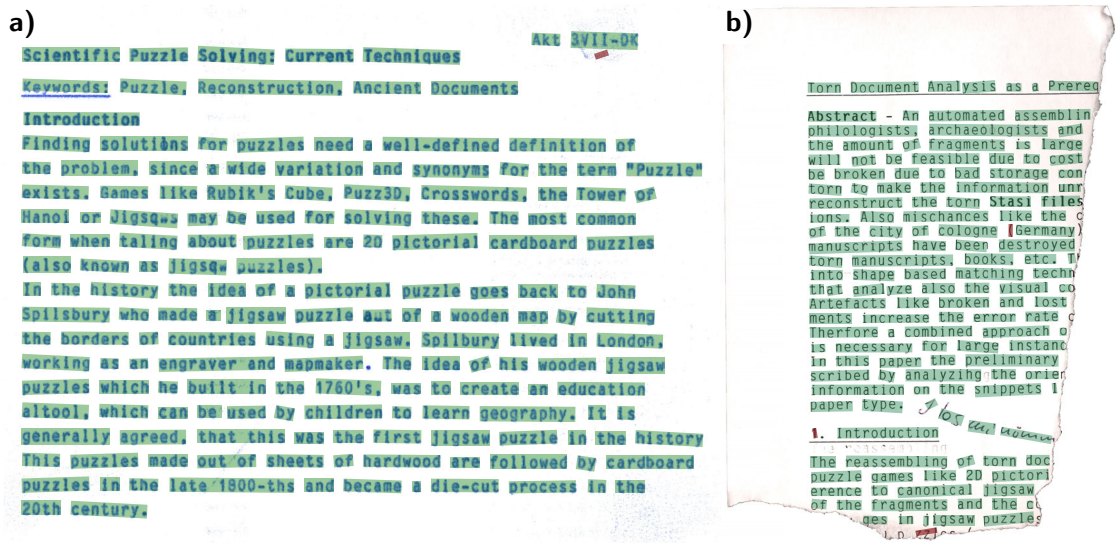


Figure 4.24: Two sample images which show some challenges of the Stasi database. Green rectangles denote **tp** while red rectangles denote **fp**.

defined as e.g. **tp** if most ground truth pixel contained in the box have the same label as the box itself. The evaluation with this metric is presented in [DKS11].

The confusion matrices in Table 4.4 and 4.5 show the class confusion if the pixel based measure and the rectangle based measure are used respectively. It is shown that the pixel based measure is generally worse than the rectangle based. This can be attributed to the fact that the rectangle based measure does not account for slight inaccuracies resulting from the prediction or the ground truth itself. In addition, for the pixel based measure the CCs need to be labeled since the class label is solely known for the profile boxes which do not necessarily correspond.

	predicted			
	noise	print	manuscript	#
noise	0.597	0.203	0.200	1 538 127
print	0.033	0.869	0.098	4 540 398
manuscript	0.013	0.039	0.948	2 935 072
	1 106 159	4 375 538	3 531 900	9 013 597

Table 4.4: Confusion matrix of the three classes if the pixel based measure is used.

Table 4.4 shows that in total 9.01 million foreground pixel are evaluated. The noise class has the worst precision of 0.597. This can be attributed to the fact that some snippets contain bleed through text. These text areas are annotated as background, since they are per definition not a readable text. Nevertheless, they have similar shapes compared to text and are therefore confused with text areas by the classifier. The table

also indicates that printed text is more likely to be confused with handwritten text than vice versa. Table 4.5 has similar relations between the class confusion. However, the overall precision is higher due to the different evaluation metric. Although the pixel wise evaluation allows for more accurate conclusions, this evaluation metric reflects the objective of the text classification better because the rectangles and text lines are used for further processing.

	predicted			
	noise	print	manuscript	#
noise	0.625	0.065	0.310	245
print	0.005	0.945	0.050	2180
manuscript	0.018	0.044	0.938	2034
	200	2166	2093	4459

Table 4.5: Confusion matrix of the three classes if the rectangle based measure is used.

In addition to the confusion matrix, Figure 4.25 gives a comparison of the precision for each class. If the box based evaluation is regarded both text classes (*printed*, *manuscript*) have similar precision. However, using the pixel based evaluation, the precision of printed text drops. The overall pixel based precision is 0.851, the recall is 0.953, the F-score is 0.899 and the accuracy is 0.835. Comparing these results to the databases previously evaluated, it can be concluded that the real world Stasi snippets are more challenging.

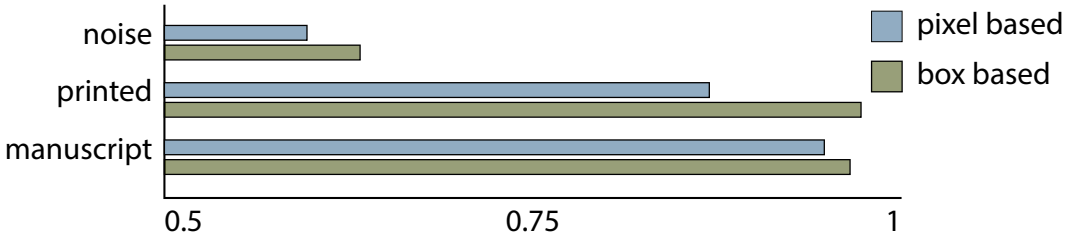


Figure 4.25: Comparison of class precision with pixel based on box based error metric.

4.5.5 Parameter Evaluation

The parameter evaluation is carried out on the training set of the Stasi database. In order to evaluate the whole pipeline with every parameter, a 3-fold cross-validation is carried out. The maximal performance of the grid search (the grid parameters are $\langle C, \gamma \rangle$) is further considered. By these means, the classifier is adopted with every parameter change which minimizes the bias introduced by the classifier.

Computing all features with each parameter value on the training set takes ages[1]. That is why, a trade-off between computation time and explanatory power of the tests

[1]The training set consists of 154 document snippets which results in 21,500 features. Hence, training the SVM once takes $\approx$ 10 min. Now if we choose a comparatively small grid with 24 tuples, the cross

needs to be performed. Therefore, the grid search is performed by selecting 2,000 feature vectors using equidistant sampling of all document snippets in the training set. Decreasing the computation time is also achieved by reducing the dataset's size. However, features within one snippet are correlated since e.g. the writing source, supporting material or the level of noise do not necessarily change.

In order to estimate the level of significance, the parameter test is performed seven times with the same parameters. Figure 4.26 shows the spreading of results with different test setups. Since the number of samples is low ($N = 7$) boxplots which are based on robust statistics measures are used for illustration. The median performance varies with the setups. For an improved comparison, the medians are aligned in Figure 4.26. The first row shows a random feature selection during cross-validation. Its interquartile range ($iql = Q_{75} - Q_{25} = 0.0067$) is close to the same setup with 500 feature samples and equidistant (*fixed*) sampling ($iql = 0.007$). The lowest interquartile range of 0.0038 is achieved with 154 test images and 2,000 feature vectors sampled at a fixed distance (4^{th} row). Hence, this setup is chosen for all subsequent parameter evaluations.

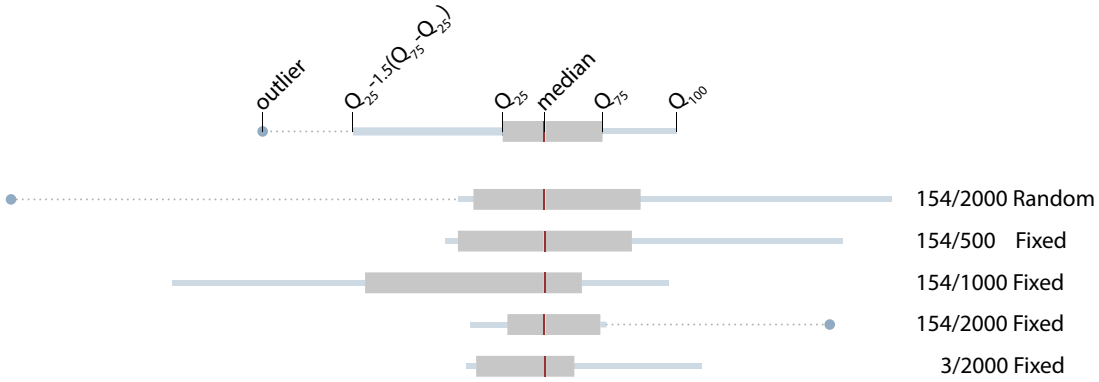


Figure 4.26: Boxplots of different test setups. The labels are *# test images/# features, sampling method*.

Multi-Angle Threshold: The multi-angle threshold T_o is used for assigning multiple orientations to a word box. If it is set to e.g. 0.6 orientations whose bins are above 60% of the maximal bin are additionally assigned to a text box. Multiple features are then computed per word, each normalized with a different angle. This strategy is inspired by the SIFT orientation normalization [Low04]. Figure 4.27 illustrates the precision with varying thresholds T_o. The maximal precision with multiple angles is achieved if $T_o = 0.5$. However, no multi angle $T_o = 1$ is best with a precision of $p = 0.898$. Since no multi angle is faster (fewer features are extracted), the multi angle normalization is not used on the Stasi dataset.

validation takes 4 h. Based on this time estimation, if 45 parameter values (this is still one test) are evaluated the test would take about a week. The feature computation itself is negligible in this scenario as it takes ≈ 5 min for all pages and parameter values.

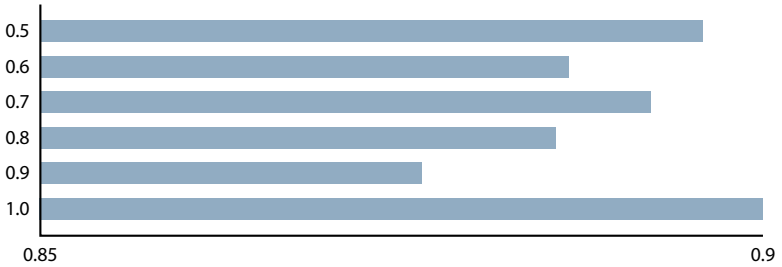


Figure 4.27: Precision when varying the multi-angle threshold T_o.

Rotation Test: A dominant orientation estimation [DKS12] is performed in order to deskew document snippets and therefore improve the performance of text classification and layout analysis. However, in the context of handwritten documents slight changes in the orientation occur if e.g. the writer does not use ruling lines. In order to test the method's robustness with respect to these variations, the dominant angle provided to the text classification module is synthetically distorted. It is varied between $(-90°, 80°]$ with a step size of 10°. In contrast to all other parameter tests, the SVM is not trained for each parameter change because the robustness is tested in this scenario.

The dominant angle changes the construction of oriented BBs and the estimation of profile boxes. Therefore, different features are extracted if the dominant angle is changed. Figure 4.28 illustrates the system's performance with synthetically distorted angle estimations. It is shown that the recall does not significantly change with increasing angle changes ($\sigma = 0.0061$). This can be attributed to the fact that the recall which captures **fn** rather depends on the segmentation than the classification. In contrast, the system's precision is decreased by 10.5% if the dominant orientation estimation error would be around 45°. The precision changes indicate that the classification performance is reduced if the features are extracted in a different (wrong) way. The slight increase of the precision around ±90° can be traced back to a higher similarity of characters which are rotated by 90° than those rotated by 45°. Figure 4.28 shows additionally, that the classification performance is robust up to ±10° where the F-score decreases by 1.97%. Note that the mean error of the orientation estimation is 0.103° [Pap+13].

CC Localization: We propose in [DKS11] to smear the binary image using LPPs. This has the benefit that fewer CCs need to be processed per page and therefore decreases computational costs. However, as discussed before, LPPs are not robust with respect to orientation changes. The parameter test performed with varying kernel size of the LPP shows that no smearing improves the classification performance. The blue (light) line in Figure 4.29 shows the precision with increasing kernel sizes. Note that the performance is monotonically decreasing (except for a filter size of 65 px) as the kernel size increases. The drawback of not using the LPP is on the one hand an increased number of CCs. On the other hand, small elements or broken characters have a lower classification performance. That is why, a second test is conducted where a morpholog-

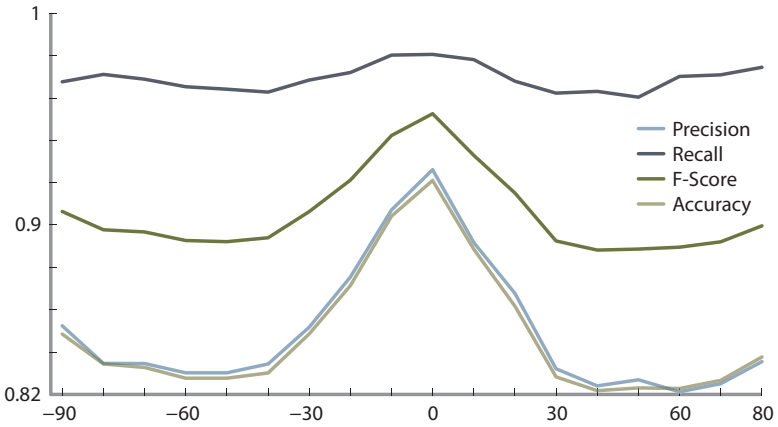


Figure 4.28: Synthetically distorted angles between $-90°$ and $80°$.

ical closing is performed on the binary image prior to the feature computation. If the structing element's radius is 3 the best performance is achieved. Note that this structing element solely bridges small gabs between CCs.

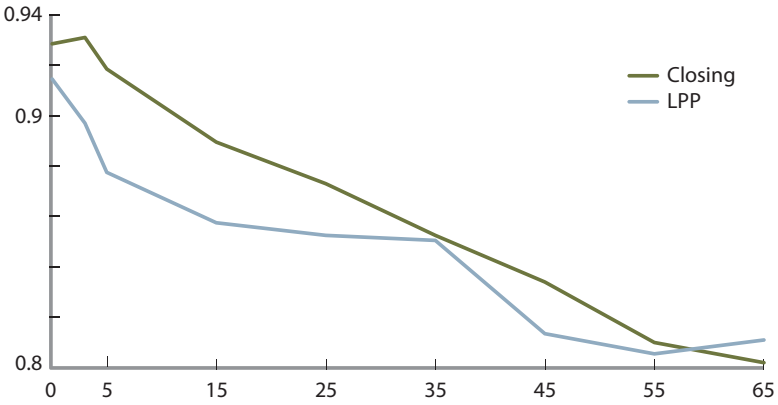


Figure 4.29: Precision with increasing kernel size of the LPP and the structing element of the morphological closing.

In addition to the CC fusion technique, three different boundary estimation methods are compared:

Minimum Area Rectangle enclosing rectangle with minimal area.

Oriented Bounding Box is a BB which is rotated with respect to the dominant orientation estimated.

Profile Box was introduced previously. It encloses a word's *x-height* rather than the whole CC.

The precision of all three methods varies by ± 0.004 between all three methods. That is why the profile boxes are used since they reduce the computational complexity. In addition profile boxes capture the local orientation of a word which improves the text line detection.

Feature Computation: The parameters which influence the feature computation are evaluated too. First, the orientation and radius sampling (θ, r) are tested. Increasing the number of radial θ and distance r bins results in a finer sampling and therefore richer representation. Figure 4.30 shows the precision when varying both parameters. The test results indicate that a coarse grid having $2 \times 2 = 4$ dimensions is not suitable with a precision being $p = 0.774$. Since the maximal performance is gained if the grid is set to $8 \times 8 = 64$ bins, this feature dimension is chosen for classification. Figure 4.30 additionally shows a performance drop if the feature dimension is further increased. Obviously the feature's distinctiveness is not further increased with increasing the feature dimension. Furthermore, slight localization inaccuracies might result in different features if a fine sampling is applied.

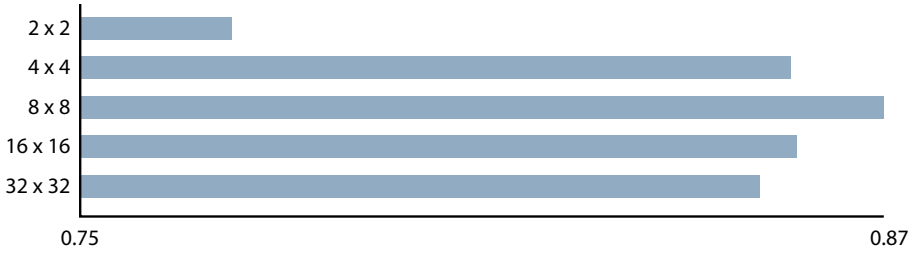


Figure 4.30: Varying the feature dimension between 4 and 1024. A 8×8 log polar grid is best suited for the text classification.

In addition to the feature dimension, the feature's size has an impact on the system's performance. It was discussed previously, that the features are extracted by means of the rectangle's height. This allows for a feature extraction which is robust with respect to scale changes as the rectangle's height depends on the character height. The test whose results are presented in Figure 4.31 shows the precision when scaling the features with respect to the height multiplied by a scale factor s. The scale factor is varied between 0.25 and 10 with a step size of 0.25. Figure 4.31 shows that the precision increases with increasing scale factors and becomes stable around 4. Since the computational expenses also increase with the scale factor, s is chosen to be 4. Note that the slight increase of 0.0025 when s is set to 7 can be neglected since the test's level of significance is 0.0038.

The σ used for Gaussian smoothing while computing the gradient vectors has an impact on the feature's granularity. If it is chosen to low, noise can impair the quality of the features. A large σ in contrast can disregard important features that are needed for classification. For evaluation, σ is varied between 0 (= no smoothing) to 4 which

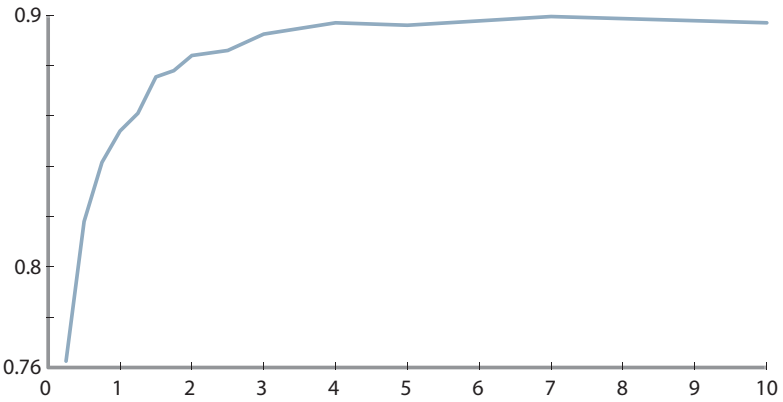


Figure 4.31: Precision if the feature's scale factor s is varied between 0.25 and 10.

would result in a kernel size of 25. Figure 4.32 illustrates the precision when σ is varied. It is shown that $\sigma = 1.0$ gives the highest precision of 0.9395. Computing the gradients without smoothing results in a precision of 0.92 which corresponds to a performance reduction of 1.95%. Additionally, the gradient magnitudes are shown if three different smoothing parameters $\sigma = 0, 1.0, 3.0$ in Figure 4.32.

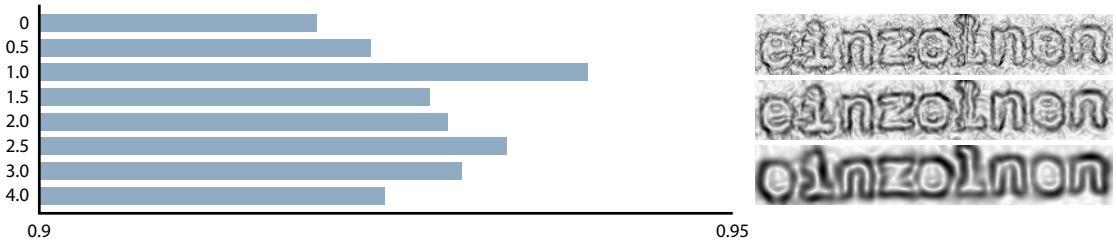


Figure 4.32: Precision if σ of the Gaussian which is needed for gradient computation is varied between 0 and 4. A word's gradient magnitude with varying σ (right).

In addition to the Gaussian smoothing parameter, the influence of the kernel size k_s which is used for weighting the gradients with the binary image is evaluated. Figure 4.33 illustrates the results of the empirical evaluation on the Stasi database. It is shown that the precision is best if k_s is set to 21. If a large kernel is applied ($k_s = 41$) the precision is minimal with 0.921. Thresholding the gradient magnitudes with the binary image ($k_s = 0$) reduces the precision too since errors in the binarization are more crucial if no weighting is applied.

4.5.6 Summary & Discussion

A new text classification methodology was presented in this chapter. The text localization is based on newly introduced rectangles which capture a word's *x-height* rather than enclosing its CC. The so-called profile boxes additionally allow for an efficient feature

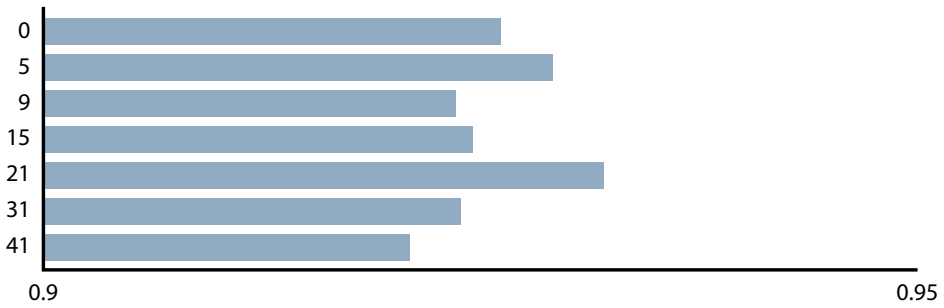


Figure 4.33: Precision when varying the kernel size k_s between 0 and 41 of the weighting image which suppresses noise.

localization. The features are extracted using a sliding window approach. GSFs are computed for each sliding window which transform the gray values into a representation which is robust with respect to noise and transformations anticipated and furthermore reduce the amount of information. A multi-class SVM assigns probability estimates for all three classes. Then, the profile boxes are labeled by voting probability estimate histograms of all features within a profile box. It was shown in the preceding section, that the classes can be varied depending on the database.

The evaluation is carried out on four different databases. The PRImA database showed the method's ability to adopt for modern newspaper articles. Instead of classifying handwritten text which is not present in this database, pictures, and diagrams are classified in this evaluation. On the IAM-DB and CVL-DB, the method's capability of dealing with different cursive handwriting is shown. Furthermore, the text classification proposed is compared to other state-of-the-art text classification approaches using the IAM-DB. The results on real world data indicate that the system is capable for automated document processing although there is still room for improvements.

CHAPTER 5

Layout Analysis

The layout analysis presented in this chapter aims at mining the knowledge of a document snippet gained during text classification to a hierarchically higher level. The layout analysis uses a bottom-up approach for grouping. The difference between top-down and bottom-up layout analysis was discussed in Chapter 2. For document snippets a bottom-up approach has the advantage that it can cope with parts of pages even if the global context of these parts is unknown. Moreover, bottom-up approaches are more flexible if e.g. handwriting intersects with printed text because there is no need for incorporating global knowledge of a document's layout.

The text line clustering is inspired by the Docstrum which is proposed by L. O'Gorman [OGo93]. Hence Profile Boxes are grouped by means of their nearest neighbor distance. In contrast to state-of-the-art text line segmentation approaches that aim at pixel accurate labeling, the approach proposed targets text line detection on the basis of rectangles. Rectangles are beneficial for reassembling document snippets because they have a simple representation and incorporate a snippet's content which allows for extrapolation at its edge. Text lines are used in order to reject possible candidates during reassembling if they have different text labels and to accurately align matching pairs based on their text lines. The next hierarchical level composes text boxes (i.e. *paragraphs*) which integrate text lines with a similar dominant orientation and a constant line spacing. Based on text boxes global attributes such as *dominant word skew, dominant text class, mean text line spacing,* etc. are extracted. The global attributes are used for searching possible matches for a document snippet. They are computed by means of bottom-up voting. A final top-down voting corrects word labels based on class probabilities of their neighbors.

The layout analysis is evaluated on three Handwriting Segmentation Contests [GSL09; GSL10; Sta+13]. In addition, the *Saint Gall* database which is composed of medieval manuscripts demonstrates the system's capability of adopting to different document analysis scenarios. The evaluation on the PRImA database which was presented in Section 4.5.1 showed the layout analysis performance on modern printed documents. However, in the context of layout analysis freestyle handwritten documents are consid-

ered as more challenging since they have local skew deviations within text lines, varying spacing, and varying word heights. That is why the evaluation in this section has a focus on layout analysis for handwritten documents.

The remainder of this chapter is organized as follows. The subsequent section details the methodology of text line clustering and the voting approaches. Then, Section 5.2 outlines the methods used for accurate text line localization. Section 5.3 discusses the results gained on Handwriting Segmentation Contest databases (Section 5.3.3) and on the medieval *Saint Gall* (Section 5.3.4) database.

5.1 Text Line Clustering

The text line clustering follows a simple design idea. It is assumed, that every Profile Box has exactly one successor. In natural language, this constraint is preexisting by definition. However, when text is analyzed automatically, the condition is not satisfied in general due to malicious Profile Boxes or background noise. The advantage of such a severe clustering rule is that false text line merges are reduced. On the opposite, a false word merging may result in text lines which are split in the middle since no proper successor can be found for a word.

The first step of clustering is the computation of nearest neighbors. Therefore, an $N \times N$ look-up table is created which stores all distances between Profile Boxes with N being the number of boxes. The distance table is not symmetric since the left-right and right-left distances are computed for each Profile Box. Computing a look-up table improves the speed since the computation of distances is carried out once with a computational complexity of $O(N^2)$. Although, the look-up table improves computation speed since each distance is computed once, it could result in memory issues. If 10,000 words are detected, a memory block of $10,000^2 \cdot 4$ bytes = 381.46 MB must be allocated. Considering a full US letter page which has the IEEE Transactions style (10 pt, two columns) that is used in conference proceedings, an average page has $\approx 1,000$ words.

Simply computing the distance between the rectangles' left and right middle points proved to result in a poor text line clustering performance if local skew is present in text lines or Profile Boxes are poorly located. That is why the minimal distance between the left and right upper, lower, and middle points is used as basis for clustering. Using the minimal distance between these three distances reduces the effect of Profile Boxes having different heights. The rectangle's "*left*" and "*right*" edges are detected with respect to the dominant orientation which allows for a text line clustering invariant with respect to the document's skew.

A potential text line fusion is illustrated in Figure 5.1. The upper left and lower right Profile Boxes have a similar word height and have the lowest Euclidean distance of 11.95 mm if the distance of the left and right middle point is regarded (gray dashed line). The upper right box is further away and larger than the left box. This error occurs if the upper profile line did not fit because of ascenders. In such a scenario the baseline of both boxes is likely to be similar as the words share the same baseline. If the minimal distance of all three rectangle points is used rather than that of the middle point, the upper line is

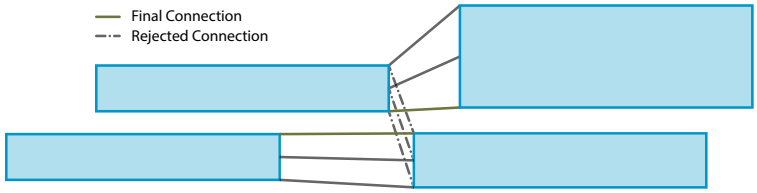


Figure 5.1: Example of possible text line merging. Using the minimal distance between the upper, lower, and middle points reduces the chance of wrong line merges. Solely the three relevant connections are illustrated for clarity.

clustered correctly because the minimal distance is 11.18 mm. If each Profile Box could have more than one successor, the upper right rectangle would merge with both other rectangles since the distance between the lower left and lower right rectangle is maximal in this sample. However, merging the upper left and upper right rectangles rejects the connection between the upper left and lower right rectangle due to the single connection rule. Both green lines denote the respective minimal distance connecting line.

In order to further minimize the chance of false vertical merges caused by low line spacing, the connecting distance $d(p, q)$ of two rectangles p and q is weighted by:

$$d(p, q) = e(p, q) \cdot (1 + C\ g_\sigma(\cos(\theta_1 - \theta_2))) \cdot (1 + C \cos\theta_c) \tag{5.1}$$

where $e(p, q)$ is the minimal Euclidean distance previously introduced and C is an empirically found constant which weights the impact of the angle penalty. It is set to $C = 15$ if the both boxes are labeled as printed and $C = 4$ otherwise. The first term penalizes rectangles with a skewed connecting line with θ_1 being the angle of the left rectangle and θ_2 the angle of the connecting line. The Gaussian distribution g_σ with $\sigma = 1/3$ penalizes connections which are orthogonal to the dominant orientation more than those which are parallel. The second term penalizes connections depending on the Profile Box location. Here, θ_c is the angle between both rectangle centers. Hence, if the angle is 0, the rectangles are aligned horizontally with respect to the dominant orientation and the connecting line is not penalized.

Figure 5.2 gives an example of a potential wrong text line merging. In this scenario, the Euclidean distance of the connecting line between "*and*" and "*region*" (dashed line) is larger than that of "*and*" and "*Ocean*". However, the weighting penalizes this connection since it detects a possible vertical line merge. In this example, the false connection would be rejected anyway since "*Indian*" is closer to "*Ocean*".

Having created the weighted distance look-up table, Profile Boxes are clustered to text lines such that the global distance is minimized. Therefore, the nearest neighbors of each word are considered. If the distance to the nearest neighbor is lower than its distance to all of its neighbors, the rectangles are connected. The merging algorithm is detailed in Algorithm 1. By these means, the graph cut problem is solved in $O(N^2)$. It is optimal in the sense that each connection has the lowest edge weighting possible.

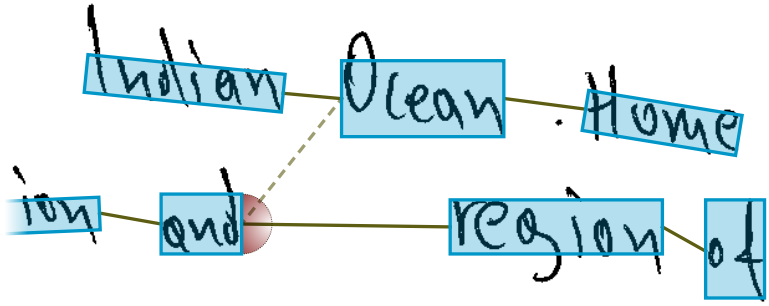


Figure 5.2: Text line merging. The red semi-circle denotes the angle weighting proposed. Darker regions indicate connecting lines which are penalized stronger. The dashed line shows a potential false connection.

In order to stop the line merging at the beginning and end of a text line, a distance threshold T_d is introduced. Depending on the application, a fixed threshold can be applied. For evaluation on the ICDAR Handwriting Segmentation Contests a fixed threshold of 500 is chosen which was found empirically. In the context of text line detection in heterogeneous documents a dynamic threshold T_d is found. The dynamic threshold is applied if more than 150 words are detected. It is defined by:

$$T_d = Q_{75} + 1.5(Q_{75} - Q_{25}) \tag{5.2}$$

where Q_{75} is the 75% quartile of the distance of all nearest neighbor connections. By these means the average word distance of a document is found robustly. In general, the *left* distance of the first word in a text line is large since *left-left* connections between words are not allowed.

In addition to the maximal distance, stopping conditions can be incorporated. Depending on the documents, either a tab stop analysis, a text line separator or both are used to stop clustering. The text line separator is typically a long vertical line which separates two or more paragraphs. Words must not be clustered to text lines if the center of gravity of one word lies on the other side of the line than that of the other word. In addition tab stops are searched for, which reduce the chance of falsely clustering two paragraphs with a low horizontal spacing. Therefore, a virtual line is detected if more than five words share common start or end coordinates in an epsilon environment.

Text Line Voting: The probability of an element for being of a specific class (e.g. *printed*) depends on its neighbors. That is why a class voting is performed based on the text line clustering. First a bottom-up voting is applied which assigns a class label to each text line based on the scores of its children. Empirical studies showed that an area weighting improves the results. Therefore, the voting can be formulated by:

$$w_{tc} = \frac{\sum_{i=0}^{N} w_{ci} \cdot a_i}{\sum_{i=0}^{N} a_i} \tag{5.3}$$

input : Distance look-up column sorted DistLookup, index look-up IndexLookup
output: Rectangle connections leftRightIdx

```
1  for rows r in DistLookup do
2  |  SortedRow ← sort(r) ;
3  |  for columns c in DistLookup do
4  |  |  leftIdx ← SortedRow at c;
5  |  |  rightIdx ← IndexLookup at leftIdx;
6  |  |  if DistLookup at r, leftIdx > T_d or leftIdx = rightIdx then
7  |  |  |  continue ;
8  |  |  end
9  |  |  if leftRightIdx at rightIdx = leftIdx then          /* No self-loops */
10 |  |  |  continue ;
11 |  |  end
12 |  |  if leftRightIdx at leftIdx has no pair and rightLeftIdx at rightIdx has no pair
   |  |  then      /* My left-right distance is smaller or equal to the
   |  |  neighbor's right-left distance */
13 |  |  |  leftRightIdx at leftIdx ← rightIdx;
14 |  |  |  rightLeftIdx at rightIdx ← leftIdx;
15 |  |  end
16 |  end
17 end
```

Algorithm 1: Text line merging

where w_{tc} is the text line weight of class c, w_{ci} is the class weight of the i^{th} class with a_i being its area in px and N the number of words in the text line currently observed. Since the classification weights of each class are propagated rather than final class labels a soft voting is performed. Therefore, words with low class probabilities (i.e. words where the classification result is doubtful) contribute less to the voting or vote for two classes at the same time. Weighting with the word's area improves the result if spurious words – which might be ascenders split by the text line detection – are present. These noisy words have in general a small area and a low classification probability since too few structures are provided during classification. Depending on the writing there might me more spurious words in a text line than "real" words. If the area weighting is applied, these words contribute less to the bottom-up voting. The class label with the maximal voting probability is assigned to each text line.

Having assigned class labels to text lines, they are further grouped to text blocks based on the median text line spacing. For each text block, the same bottom-up voting is applied for text blocks and finally for a whole document page. The global class probabilities for a snippet are used in order to group them before reassembling.

The class labels of hierarchically higher order elements such as text blocks and text lines are used for a neighborhood voting of words. Therefore, a scheme similar to the

bottom-up voting is applied in reverse order. Hence, a top-down voting is applied to text blocks in order to correct for false text line labels. The mean class probability between a parent's element and its child is assigned to all children. If the maximal bin is changed by these means, a new class label is assigned to the child. The advantage of the top-down voting, is that the voting incorporates context of text elements found in the bottom-up clustering approach. Therefore, a text box has less influence on a word than its text line which is hierarchically closer to the word layer. By computing the mean class probability solely words with low probabilities (i.e. doubtful decisions) are relabeled. Hence, handwritten words with a high accuracy which are surrounded by printed text are not relabeled while e.g. noisy words with a low accuracy are corrected based on their neighboring words. By these means, the voting improves the classification precision while still accounting for mixed text environments such as forms.

Noise located at the border of documents decreases the recall of text lines since they grow too large if these elements are grouped with text elements. Therefore, Profile Boxes which are labeled as noise after the top-down voting are filtered. Then, the text line clustering is performed again so as to guarantee a clustering step on "clean" data.

5.2 Text Line Localization

Text lines are used during reassembling to reject wrong candidates and for fine tuning the location of snippet pairs. In contrast to state-of-the-art methods such as the participants of the ICDAR Handwriting Segmentation Contests [Sta+13], the proposed methodology has no need for pixel accurate text line labeling. One reason is that pixel are more complex to describe and store even if their contours are implicitly used as abstraction layer. The other reason is that pixel at the border of snippets may be noisy because of the edge rupture or ascenders and descenders close to edges.

Hence, a representation is needed which incorporates the knowledge of text lines and can accurately localize text lines even at the edges of snippets. Therefore, rectangles are used to represent text lines. They have the advantage of a simple data representation since solely five floating point numbers are needed (center, size, and angle). Furthermore, text lines may exhibit changing skew deviations. However, the changing skew within text lines can be neglected. Figure 5.3 shows a detail of a page from the ICDAR 2013 Handwriting Segmentation Contest [Sta+13]. The text line localization adopts to the local skew of text lines while being robust with respect to skews of single words (e.g. "*Washington*" at the beginning of the second last text line).

The robust fitting of text line rectangles with respect to Profile Boxes which are extracted during text classification is achieved by means of the PCA. First points are sampled equidistantly along the upper and lower lines of rectangles. The point sampling weights rectangles with respect to their width. Therefore, long rectangles contribute more than short ones. Then, the text line's angle is estimated by applying the PCA to the upper and lower sample points. The text line angle is therefore defined by:

$$\theta_t = \tan^{-1}\left(\frac{e_1}{e_2}\right) \tag{5.4}$$

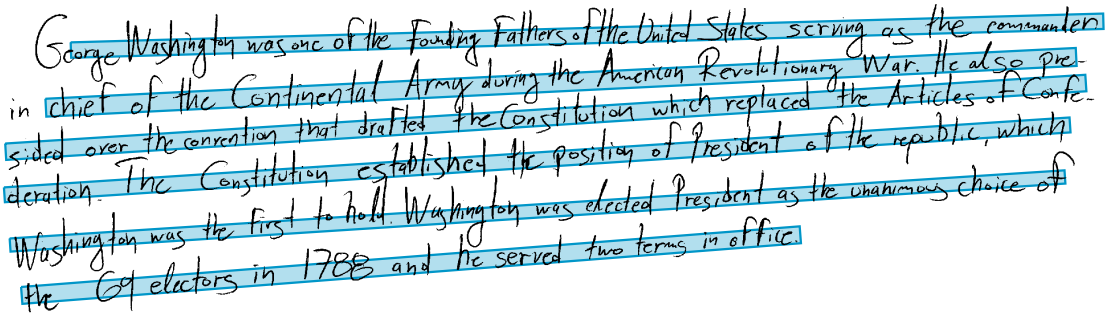


Figure 5.3: Text line localization on a sample snippet of the ICDAR 2013 Handwriting Segmentation Contest.

where θ_t is the resulting dominant orientation of a text line and e_1, e_2 are the first and second Eigenvectors respectively. The dominant angle is robustly found with respect to outliers by these means. Then, the text line's height is defined as the difference between the median of the upper and lower sample points with respect to the dominant orientation θ_t. Computing a robust statistic allows for a compact text line representation even for short text lines when outliers such as ascender/descender boxes are present. Figure 5.4 shows the text line fitting on a page from the ICFHR 2010 Handwriting Segmentation Contest [GSL10]. The PCA angle estimation determines the dominant angle correctly even if noisy elements are present at the beginning or end of a text line (e.g. the split text box above the Π at the beginning of the last text line).

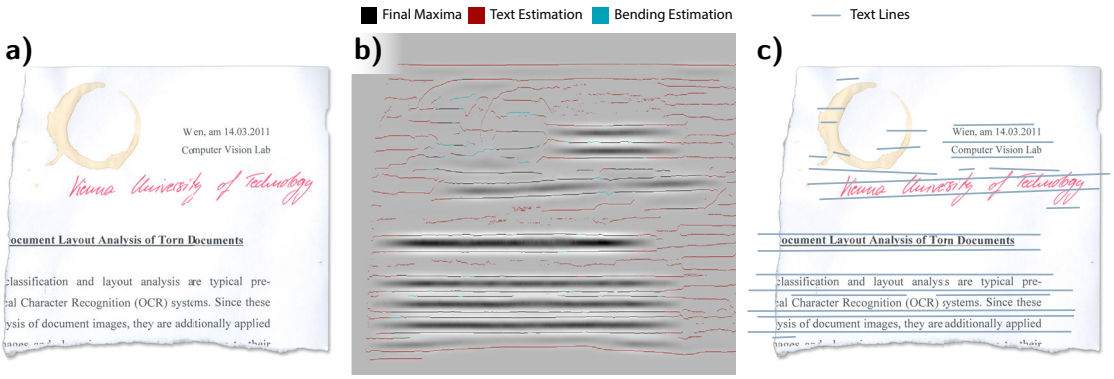


Figure 5.4: A sample page from the ICFHR 2010 Handwriting Segmentation Contest with Profile Boxes and the text line rectangles fit.

Pixel Accurate Segmentation: The pixel accurate text line segmentation is needed to compare the methodology proposed with state-of-the-art handwriting text line segmentation algorithms. In order to map the segmentation result to pixel Profile Boxes are rendered to the images. For each Profile Box the line index is stored in the image.

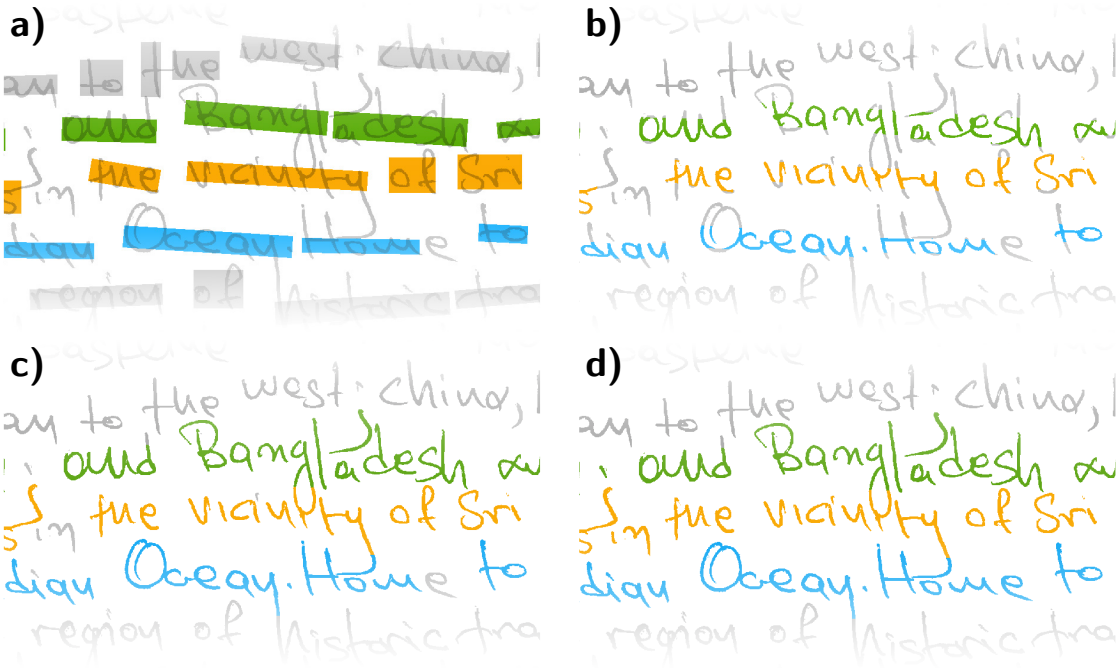


Figure 5.5: Text line labeling. Binary image with clustered Profile Boxes a), Indexes combined with the binary image b), labeling after region growing on CCs c) and final labeling result after assigning labels to missing CCs in d).

The thus rendered image is combined with the binary images by means of a logical AND operation. Then a region growing is performed which assigns the label index to unlabeled parts of CCs. By these means unlabeled pixel of CCs with more than one line index get the index of the text line which is closer. In addition, CCs which are not covered by a Profile Box (e.g. CCs with an area which is too small) get the line index of the text line which is closest. The metric for determining the closest text line prefers horizontal distances rather than vertical distances.

The labeling procedure is shown in Figure 5.5. Each text line index is illustrated by different colors. First the Profile Boxes are rendered to the image in a). The second image b) shows the combination of the rectangle image with the binary image. Note that some parts of CCs are not labeled yet. After region growing (c) all CCs covered by rectangles are indexed. The "*g*" in *Bangladesh* has two line indexes. However, the region growing does not split the CC optimally. After region growing CCs which were not covered by any rectangle are not labeled (e.g. the "*e*" in *Home*). These CCs are labeled in the final step (d) by grouping them to the text line which is closest.

5.3 Evaluation

The evaluation of the layout analysis is carried out on publicly available databases. It is not evaluated on the Stasi database introduced previously because the ground truth for layout analysis is missing. The databases which are detailed subsequently contain handwritten documents. Note that the layout analysis system is not evaluated on printed data because printed text is in general perfectly aligned. Hence, there is no need for sophisticated layout analysis systems if printed text is observed. Nevertheless, the evaluation on the PRImA database in Section 4.5.1 demonstrated the system's capability of dealing with printed pages with various non-Manhattan layouts. The databases used for evaluation are subsequently listed.

ICDAR *2009 Handwriting Segmentation Contest*
This database contains 200 handwritten document images (4034 text lines) written in English, French, German, and Greek [GSL09]. All images are binarized and do not contain noise or non-text elements.

ICFHR *2010 Handwriting Segmentation Contest*
This database is similar to the ICDAR 2009 Handwriting Segmentation Contest with 100 images (1,629 text lines) [GSL10]. However, more overlapping and slanted text lines are present which increases the difficulty.

ICDAR *2013 Handwriting Segmentation Contest*
Again, 150 handwritten document images (2,649 text lines) are used [Sta+13]. In this contest, non-Latin scripts are such as Bangla are introduced. The difficulty of Bangla are accents above and below text lines.

Saint Gall Database
The database consists of 30 medieval manuscript pages which are composed of 720 text lines [Fis+11]. Pages of this database have stains, holes, and notes beside the actual text. It is shown in Figure 5.6 that the writing differs substantially from modern handwriting. Additionally, initials which span 2-3 text lines complicate the text line detection.

Figure 5.6 illustrates a detail of a page from the *Saint Gall* database. Note that the handwriting is – similar to printed text – aligned with short and in most cases non-overlapping ascenders and descenders. Here, the challenge is on the one hand to correctly classify into text and non-text elements. It is shown in Figure 5.6 a) that the red initial is correctly labeled as background element (gray and yellow). In addition, annotations close to paragraphs complicate the text line extraction.

A sample page of the ICDAR 2013 Handwriting Segmentation Contest [Sta+13] is shown in Figure 5.7. The page is written in Bangla and has text lines with a slight local skew. In Figure 5.7 b) the text lines automatically extracted are shown. Note that the rectangles adopt to a line's local skew. The final result which is a labeled binary image is given in c). Pixel with the same color belong to the same text line. Some ascenders, descenders, and accents are falsely labeled between line 13 and 14.

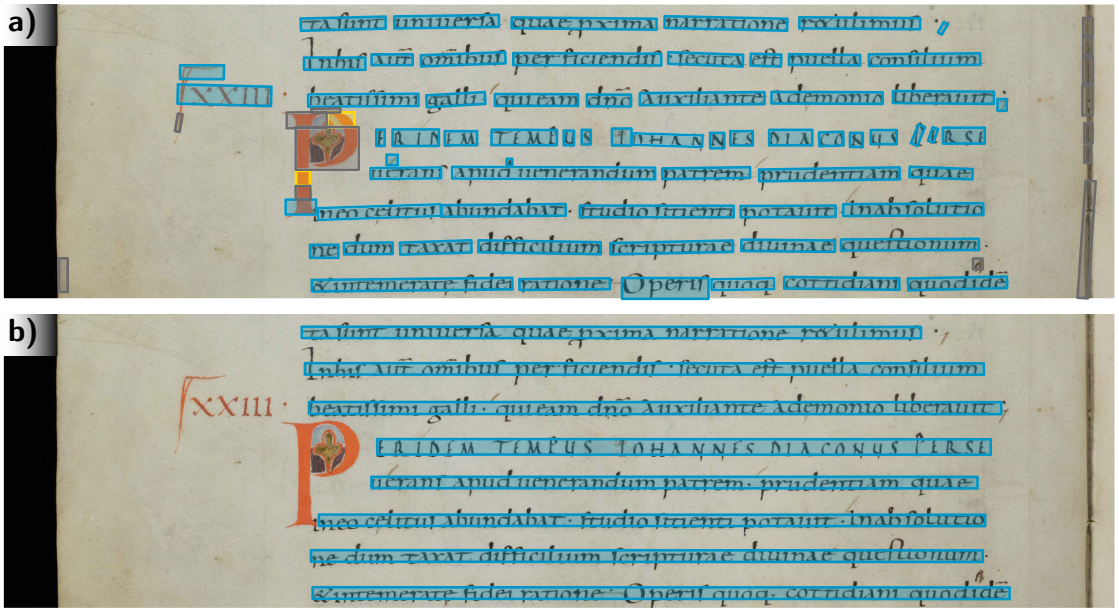


Figure 5.6: Detail of a page from the *Saint Gall* database. The initial is correctly rejected because of the text classification. Resulting text lines b).

5.3.1 Text Label Voting Evaluation

Before going into detail of the text line evaluation, the performance of voting is analyzed. The text classification was discussed in Chapter 4. The system's accuracy is improved, if words are relabeled with respect to their neighbors' labels. This performance improvement is evaluated on the Stasi database which was introduced previously. Table 5.1 shows the class confusion if a local neighborhood voting is applied with respect to the text line segmentation. The voting especially improves the classification of printed text. This can be attributed to the fact that the intra-class variability of printed text is lower than that of *noise* or *manuscript*. In contrast, the recognition of *noise* is reduced when local neighborhood voting is applied.

	predicted			
	noise	print	manuscript	#
noise	0.462	0.296	0.243	1 040 668
print	0.005	0.904	0.096	4 386 240
manuscript	0.005	0.056	0.939	2 901 968
	497 425	4 434 708	3 396 743	8 328 876

Table 5.1: Confusion matrix of the three classes if voting is applied.

In order to demonstrate the effect of voting if document snippets are analyzed, the per pixel precision, recall, F-score, and accuracy are compared in Figure 5.8. The pre-

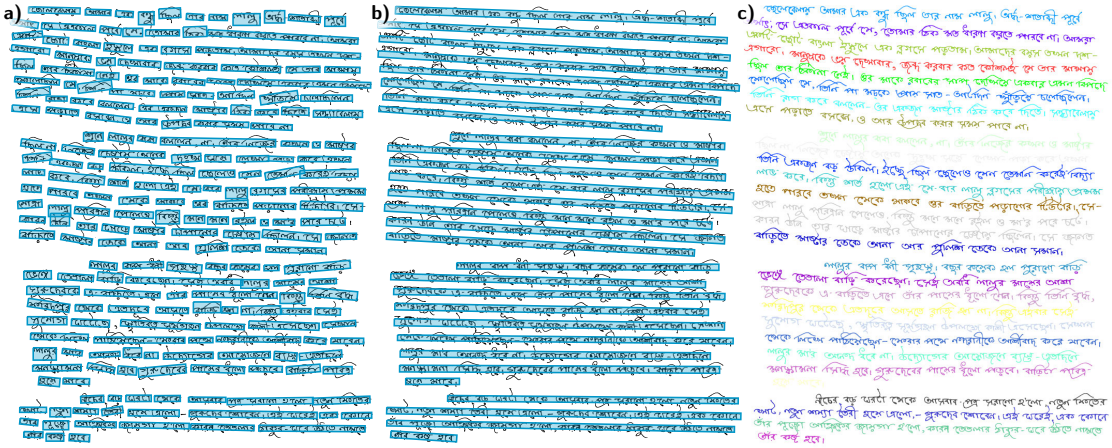


Figure 5.7: Word boxes of a sample page from the ICDAR 2013 Handwriting Segmentation Contest. The words detected a), text lines b) and the final pixel accurate annotation c). Some accents and ascenders/descenders are labeled falsely in c) (e.g. line 13).

cision is increased to 0.854 because more printed words are recognized correctly (90.4% instead of 87% without voting). While this increases the F-score at the same time to 0.8998, the overall accuracy is decreased being 0.838. This can be traced back to the decreased recognition rate of noise since **tn** are regarded for the accuracy computation but disregarded when computing the F-score. To summarize, voting improves the overall F-score by 0.05% if document snippets are analyzed. Since they have fewer text elements than full pages, the impact of class voting is comparatively low.

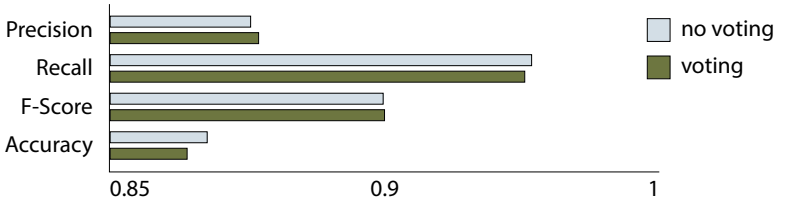


Figure 5.8: Precision, recall, F-Score, and accuracy on the Stasi database if no voting and the voting proposed are applied. The precision is slightly increased while the **tn** rate is decreased.

5.3.2 Evaluation Metric for the Handwriting Segmentation Contests

The performance metric on all ICDAR Handwriting Segmentation Contests is based on a MatchScore [Sta+13] that computes the maximum overlap of a text region with the ground truth region. The regions are in both cases the *on* pixel of text. Hence, background is not evaluated. If this score is above a given threshold T_α which is set to 95%,

the text line is considered as correct (o2o). In other words, if more than 5% of the pixel in a text line have a wrong label, the text line is considered as wrong. Based on this MatchScore, the Detection Rate (DR), Recognition Accuracy (RA), and the Performance Metric (FM) are computed:

$$DR = \frac{o2o}{N}, \quad RA = \frac{o2o}{M}, \quad FM = \frac{2\ DR\ RA}{DR + RA} \tag{5.5}$$

where N is the number of ground truth text lines and M the number of elements retrieved by the algorithm. In other words, the DR can be considered as recall and the RA as precision. In contrast to the evaluation on the IAM-DB database in Section 4.5.2, these error metrics are the same as those proposed in the Handwriting Segmentation Contests which allows for directly comparing the proposed system with state-of-the-art text line segmentation algorithms.

Since the layout analysis system aims at detecting text lines (size, angle, location) rather than perfectly segmenting binarized text, a labeling strategy that maps the rectangles to binarized text is developed. Therefore, the word rectangles of each text line are mapped to the binary image with unique line IDs. A region growing within each CC labels all pixel of components which are covered by word rectangles. The growing guarantees, that CCs with more than one label (e.g. if the ascenders and descenders of two text lines are merged in the binary image) are correctly labeled. In addition, CCs that do not overlap with a word rectangle such as accents are assigned to the text line of the closest CC. However, this labeling is by far not optimal for such a text line segmentation scenario. Figure 5.9 shows two examples of labeling errors. While the "f" in the second line of a) is correctly labeled, the "g" in the third text line is partially falsely labeled. The second example shows the ground truth in b) and the resulting text line annotation of the proposed system. Here, both connections of the second and third text line are falsely labeled. If a text line has more malicious strokes, it is likely that the region overlap is below the threshold $T_\alpha = 95\%$. Hence such a text line is considered as false positive.

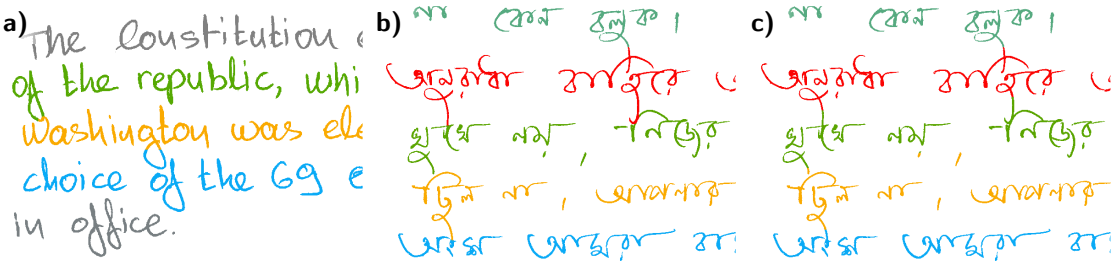


Figure 5.9: Two details of the ICDAR 2013 Handwriting Segmentation Contest. The text lines labeled automatically are illustrated in a) and c). The ground truth of c) is shown in b). Note that especially connections between text lines are labeled falsely.

5.3.3 Handwriting Segmentation Contest

The results of the layout analysis method proposed on the last three Handwriting Segmentation Contests are given in Table 5.2 and 5.3. The ICDAR 2009 Handwriting Segmentation Contest has 11 participants, while the ICFHR 2010 solely has 6 participants. The last contest at ICDAR 2013 has again 11 participants. In addition, three state-of-the-art methods are evaluated in this contest. The CUBS method [GSL10] performed best in the former two contests. However, it is outperformed in the most recent contest by INMC which is proposed by H. Koo and N. Cho [KC10] who detect lines by means of an energy minimization framework (see Section 2.4). All three tables give the number of text lines detected M, the number of text lines whose overlap is greater than T_α (o2o) and the resulting error measures DR, RA, FM.

	M	o2o	DR	RA	FM
CUBS	4,036	4,016	99.55	99.50	99.53
ILSP-LWSeg-09	4,043	4,000	99.16	98.94	99.05
CVL	4,034	3,977	98.59	98.59	98.59
PAIS	4,031	3,973	98.49	98.56	98.52
CMM	4,044	3,975	98.54	98.29	98.42
CASIA-MSTSeg	4,049	3,867	95.86	95.51	95.68
PortoUniv	4,028	3,811	94.47	94.61	94.54
PPSL	4,084	3,792	94.00	92.85	93.42
LRDE	4,423	3,901	96.70	88.20	92.25
JadavpurUniv	4,075	3,541	87.78	86.90	87.34
ETS	4,033	3,496	86.66	86.68	86.67
AegeanUniv	4,054	3,130	77.59	77.21	77.40

	M	o2o	DR	RA	FM
CUBS	1,626	1,589	97.54	97.72	97.63
NifiSoft	1,634	1,589	97.54	97.25	97.40
CVL	1,633	1,583	97.18	96.94	97.06
IRISA	1,636	1,578	96.87	96.45	96.66
ILSP-a	1,656	1,567	96.19	94.63	95.40
ILSP-b	1,655	1,559	95.70	94.20	94.95
TEI	1,637	1,549	95.09	94.62	94.86

Table 5.2: Results of the ICDAR 2009 (left) [GSL09] and ICFHR 2010 (right) [GSL10] Handwriting Segmentation Contest.

Table 5.2 shows that the proposed method (CVL-DB) is ranked third in both, the ICDAR 2009 and the ICFHR 2010 Handwriting Segmentation Contest with an FM of 98.59% and 97.06% respectively. The overall FM is lower for the ICFHR 2010 contest since the database is more challenging. Figure 5.10 shows two sample pages of the ICDAR 2009 a) and the ICFHR 2010 b) contests. It is shown in a) that the lines have solely a few overlapping CCs which eases the splitting of text lines. The sample page from the ICFHR 2010 contest c) has text lines with different sizes, local skew and overlapping components. Figure 5.10 b) and d) show the corresponding pixel mapping. It can be seen that "*Bounded*" (4^{th} line in c)) is not detected during line extraction. However, the labeling corrects this error since the fourth text line is closest to the word. Except for this error, all text lines are correctly extracted and adopted to the local skew. The FM of the page illustrated in c) is 63.16% by reason of falsely labeled ascenders and descenders especially in the first four text lines.

The ICDAR 2013 competition introduces non-Latin scripts. It was mentioned previously, that text written in Bangla is especially challenging because of the accents which are likely labeled wrong by the proposed method. In order to demonstrate this effect, the results of the proposed system are compared for each script (Greek, English, and

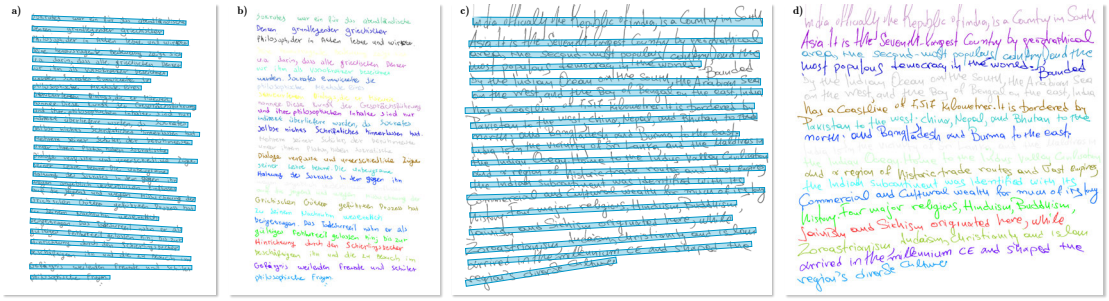


Figure 5.10: Sample images from the ICDAR 2009 (a, b) and ICFHR 2010 (c, d) datasets. The missing word *"Bounded"* in c) gets corrected during labeling d). The FM of c) is 63.16% by reason of falsely labeled ascenders and descenders especially in the first four text lines d).

Bangla). Figure 5.11 shows the error metrics for all scripts. The results are not biased since the number of documents per language is equal (50). The proposed layout analysis system has the worst performance on Bangla with an FM of 94.08%. In order to improve the text line segmentation for this script, the labeling strategy needs to be more sophisticated. Another interesting observation can be made regarding the test on different languages. Although, the system is designed for German text which use the same script as English (96.65%), the system has a higher performance on documents written in Greek (98.82%).

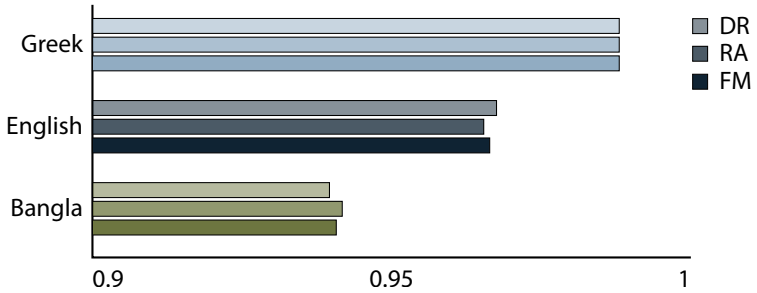


Figure 5.11: DR, RA, and FM of the proposed method on different scripts present in the ICDAR 2013 database.

Comparing the performance with respect to different languages with state-of-the-art is not possible, since it is not included in the ICDAR 2013 Handwriting Segmentation Competition [Sta+13]. Nevertheless, the overall performance of the proposed layout analysis system can be compared to 13 state-of-the-art methodologies. Table 5.3 gives a detailed view on the resulting errors. The proposed methodology achieves the 8^{th} rank. An overall FM of 96.64% is gained on this database. Since the system makes solely a few real text line errors (the FM is 99.2% if $T_\alpha = 90\%$), the labeling is currently the bottleneck for pixel accurate text line segmentation. However, it was mentioned in the

introduction that the proposed system has a focus on estimating text lines for further processing steps and alignments which have no need for pixel accurate mapping. That is why, the performance is fair enough for this application.

	M	o2o	DR	RA	FM
INMC	2,650	2,614	98.68	98.64	98.66
NUS	2,645	2,605	98.34	98.49	98.41
GOLESTAN-a,b	2,646	2,602	98.23	98.34	98.28
CUBS	2,677	2,595	97.96	96.94	97.45
IRISA	2,674	2,592	97.85	96.93	97.39
TEI*	2,675	2,590	97.77	96.82	97.30
LRDE	2,632	2,568	96.94	97.57	97.25
CVL	2,649	2,556	96.64	96.64	96.64
ILSP*	2,685	2,546	96.11	94.82	95.46
NCSR*	2,646	2,447	92.37	92.48	92.43
QATAR-b	2,609	2,430	91.73	93.14	92.43
QATAR-a	2,626	2,404	90.75	91.55	91.15
MSHK	2,696	2,428	91.66	90.06	90.85
CVC	2,715	2,418	91.28	89.06	90.16

Table 5.3: ICDAR 2013 Handwriting Segmentation Contest [Sta+13]. State-of-the-art algorithms are denoted by *.

Figure 5.12 visually compares the results of all Handwriting Segmentation Contests used for evaluation. The proposed method performs similarly to the first five participating methods in the ICDAR 2009 contest. In the ICFHR 2010 contest, the first four methods have a similar performance. In contrast, the first three methods have a similar good performance in the last contest. Here, the performance of the proposed method is in the third block with the ILSP method [Pap+10].

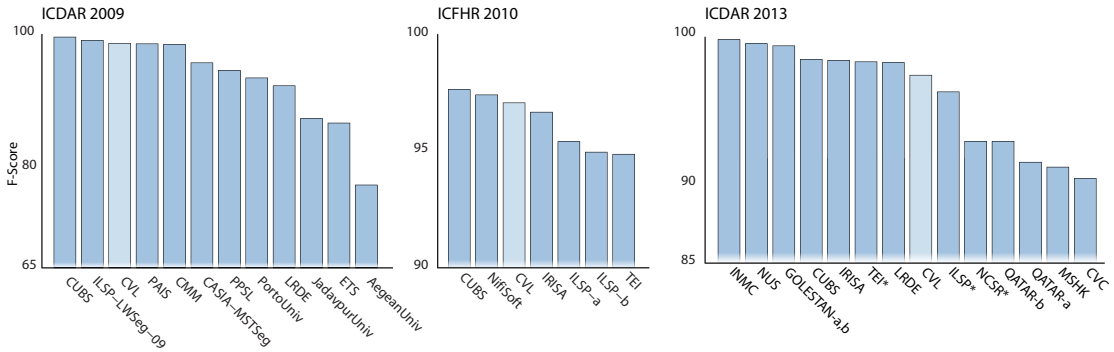


Figure 5.12: Results of the ICDAR 2009, the ICFHR 2010 and the ICDAR 2013 Handwriting Segmentation Contest.

In order to demonstrate the effect of the region overlap threshold T_α on the evaluation results, it is varied between 0.9 and 0.97. All evaluations presented previously are carried out with $T_\alpha = 0.95$ which complies with the contest evaluation. In Figure 5.13 the effect of T_α is demonstrated visually. Intuitively, the system has the maximum performance if T_α is chosen low with 0.9. With this threshold an FM = 99.2% is gained. More specifically, 21 text lines out of 2649 are wrong. An FM above 98% is maintained until $T_\alpha = 0.93$. Then, the performance drops until it reaches its minimum of 90.82% at $T_\alpha = 0.97$. With respect to this evaluation, improving the labeling procedure suggests itself since the text line localization gains sufficient performance.

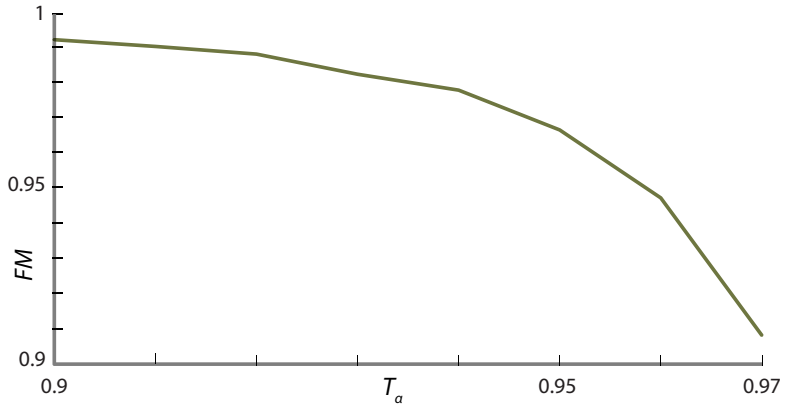


Figure 5.13: FM of the proposed system when varying the region overlap threshold T_α.

5.3.4 Evaluation on the St. Gall Database

The evaluation on the *Saint Gall* database is performed to show the system's capability of adopting to different document types. Figure 5.14 shows a detail of the database with bleed-through of an illustration from the reverse side of the page. Though, most of the bleed-through elements are falsely binarized, the text classification is capable of rejecting these elements. Note that some Profile Boxes are not properly aligned with the words in the noisy area. Nevertheless, the text line localization compensates these errors. Finally, all text lines are located correctly. In order to deal with these challenges, the classifier is trained on *text, initials,* and *noise* (rather than *manuscript*, *printed*, and *noise*) using 20 pages from the training set.

For the evaluation on the *Saint Gall* database, the error metric is slightly different to that of the Handwriting Segmentation Contests. It is adopted such that the performance can be compared to other state-of-the-art methods which do not use the error metric previously introduced. Here, the Pixel-Level Hit Rate (PHR) and the FM (also called *Line Accuracy Measure*) are used for evaluation. They were previously used for a text line detection evaluation on the *Saint Gall* database by Garz et al. [Gar+12]. The PHR is the number of pixel labeled correctly divided by the number of ground truth pixel. The threshold T_α is set to 90% in order to allow for direct comparisons. The advantage

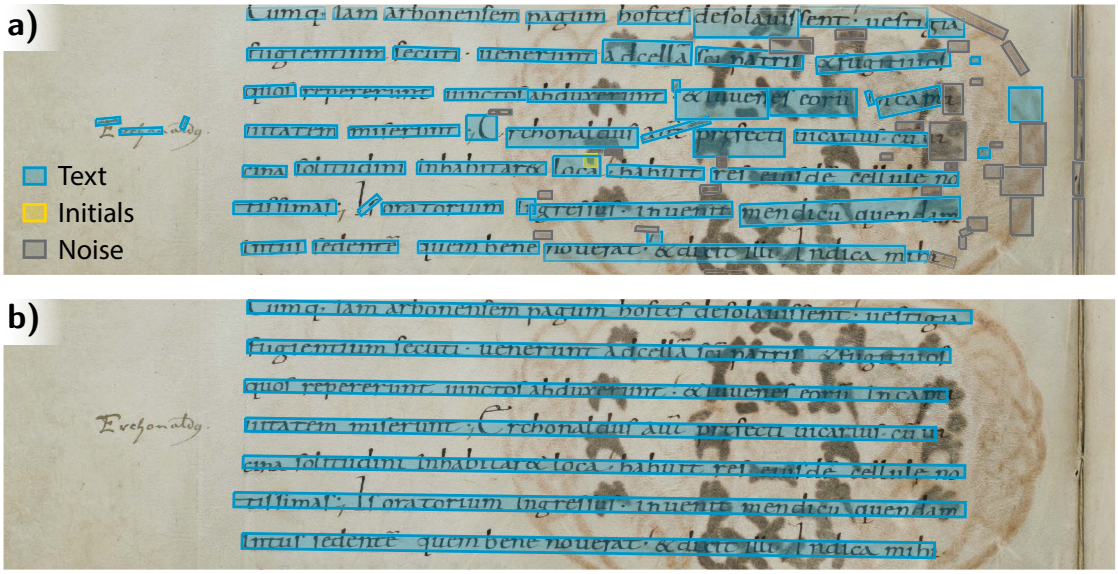


Figure 5.14: Text line localization on the *Saint Gall* database. The text lines are correctly localized even if Profile Boxes are distorted.

	# text lines	PHR	FM
CVL-DB	720	0.989	0.990
Garz et al. [Gar+12]	1431	0.987	0.979

Table 5.4: Results on the *Saint Gall* database compared to a state-of-the-art text line detection.

of the PHR in combination with the FM is that it allows for drawing conclusions about the cause of errors. Hence, if the PHR is higher than the FM short lines are more likely misclassified than long lines. Note that this argument is only valid if the area of text lines is similar.

Table 5.4 compares the results gained by the system proposed with a state-of-the-art text line segmentation. The method proposed by Garz et al. [Gar+12] was detailed in Section 2.5.3. The system achieves an FM of 0.99 which implies that solely seven lines have a region overlap below $T_\alpha = 90\%$. Compared to the binarization free text line segmentation, the PHR is not significantly better, however the FM is improved by 1%. This indicates, that the proposed system is more accurate in text line localization even if noise is present. The smaller amount of text lines evaluated can be attributed to the fact that the text classification was trained on the official training set of the *Saint Gall* database.

Summary & Discussion

Text line clustering and layout analysis was presented in this section. Text lines are represented by oriented rectangles which estimate a text line's local skew. It is shown that the dominant text line orientation is determined correctly even if noisy rectangles are present. The text line clustering proposed can be easily adopted to different types of documents. It has a low computational complexity since Profile Boxes are the basis for clustering rather than contours or pixel values. Though the graph-cut algorithm is not competitive with state-of-the-art graph-cut algorithms such as the Probabilistic Graphlet Cut presented by Zhang et al. [Zha+13], it was shown in the evaluation that the text line segmentation can compete with state-of-the-art methods. This can be traced back to the fact that knowledge about the composition of text – such as the weight of the connecting angle – is incorporated.

The evaluation on the databases of the Handwriting Segmentation Contests allowed for direct comparisons between the approach proposed and state-of-the-art text line segmentation approaches. The approach could compete with participants of the ICDAR 2009 and ICFHR 2010 Handwriting Segmentation Contests. In the ICDAR 2013 contest solely the 8^{th} rank was achieved. The recognition rate which is inferior to the best text line segmentation methods can be attributed to two circumstances. First, text lines are localized by means of oriented rectangles which give a good approximation but keep a need for a labeling strategy since the per pixel segmentation is not known. Because the system is not designed for a pixel accurate segmentation, the labeling strategy does not incorporate a-priory knowledge of text elements such as rules for optimal ascender/descender splitting or grouping of accents. This design lack (which solely becomes relevant on the contests) especially decreases the performance if non-Latin scripts are analyzed. Second, the system is in general designed to have a low computational complexity as it is applied for the reconstruction of millions of document snippets. Therefore, a global energy minimization on the basis of "characters" which is used by Koo et al. [KC12] for example cannot be applied.

CHAPTER 6

Conclusion

Document analysis applied to manually torn document snippets was presented in this thesis. After discussing related work, ruling analysis focusing on supporting material classification and accurate ruling line localization was presented in Chapter 2. A thorough evaluation on real world Stasi data and synthetically generated ruling pages allows for determining assets and drawbacks of the ruling analysis developed in this thesis. The text classification which aims at analyzing text present in documents shows that Profile Boxes improve the localization. In addition new Gsf features are introduced which prove to be designed such that they can be employed in various document analysis scenarios. The layout analysis which is a flexible bottom-up approach is capable of correctly separating text lines even if sparse data is present. All methods presented is discussed in more detail subsequently.

The ruling analysis design which first classifies documents into *void*, *lined*, or *checked* proved to have advantages compared to approaches who are directly applied on binarized lines. Ruling lines have – on purpose – low contrast which renders binarization of complete ruling lines challenging and in some cases impossible. The preceding classification step allows for designing a consecutive line localization methodology which is sensitive to low contrast lines because the information if a page is ruled or not is known at this stage. Furthermore, the line localization needs not to be performed on *void* pages which improves the computational speed. The classification assigns a label to document snippets and retrieves the dominant orientation of ruling lines. Therefore, the line localization needs not to be robust with respect to rotation which reduces the complexity and increases the computation speed. The parameter evaluation surfaced that the interpolation level used for extracting the final feature from the polar transformed power spectrum does not significantly influence the classification performance. All other parameters are crucial because they either influence the computational speed or performance of classification. The evaluation on a database synthetically generated showed that the ruling analysis is capable of removing 93% of the ruling line pixel. This result outperforms other state-of-the-art methods such as the one presented by W. Abd-

Almageed et al. [AKD09]. A drawback of the ruling line classification is its sensitivity to binarization errors. Hence, if too much foreground pixel are missed during binarization, the contrast enhancement fails which degrades the polar spectrum feature. In addition, if the binarization labels too many ruling lines as foreground elements, they are removed and therefore not visible to the feature extraction process. In both cases, the supporting material is likely to be misclassified as *void*. The latter scenario is however needed to distinguish between ruling lines and lines of tables since the only difference between these two types of lines is their contrast if document snippets are analyzed who might be fully covered by a table. Ruling location is carried out robust with respect to the dominant orientation based on the angle found during classification. If this angle is wrong, the localization is not able to recover ruling lines correctly.

Text localization which is based on binary images and robust with respect to local skews is represented by Profile Boxes. These boxes proved to have a good trade-off between discriminative power and complexity. Since rectangles are used rather than contours, subsequent processing steps are simplified which speeds up the processing chain. For short words such as "*of*" the Profile Box extraction is not applicable since too few data is present for estimating the upper and lower profile lines. In these cases a fallback to the oriented bounding box is applied which guarantees that the feature extraction and classification is improved for such scenarios. However, oriented bounding boxes decrease the layout analysis performance if their height is by far larger than that of Profile Boxes which might occur in the context of cursive handwriting.

Evaluations on the Stasi database show that the GSFs are best normalized by the word's skew angle that is computed by means of a local gradient histogram. This orientation normalization is robust with slight variations of Profile Boxes or minimum area rectangles. Synthetically introducing a dominant angle estimation error in Section 4.5.5 showed that the text classification methodology proposed is robust with deviations up to $\pm 10^\circ$. This is sufficient since the skew estimation used has an average error of 0.1° which is achieved on the benchmarking database of the ICDAR 2013 Document Image Skew Estimation Contest [Pap+13].

The GSFs proposed are evaluated on the benchmarking database of the ICDAR 2009 Page Segmentation Competition [Ant+09a] where an F-score of 0.9447 is achieved. This result outperforms participating methods. This evaluation showed that the system proposed is particularly good at recognizing non-text elements which can be attributed to the rejection of noise elements and the capability of training new object categories such as graphics or line drawings. On the IAM-DB which mainly consists of cursive handwriting, recognition scores are achieved which can compete with state-of-the-art methods. In contrast to other methods, all 1,534 pages are used for evaluation. Furthermore, this database is exhausted in the context of text classification since recognition accuracies of 99.8% are achieved.

Layout analysis is carried out on labeled Profile Boxes. By these means a global energy minimization can be computed sufficiently fast. The evaluation on three of the most recent Handwriting Segmentation Contests [GSL09; GSL10; Ant+13] showed that the layout analysis can challenge state-of-the-art methods. In this analysis scenario, the

major disadvantage of the methodology proposed is its lack in labeling text lines pixel accurate. Especially Bangla script could not be segmented with sufficient accuracy. An improved labeling strategy and incorporating a-priori knowledge of clustering accents would enhance the system's results. An evaluation of the system's performance on the *Saint Gall* database demonstrated its flexibility and ability to adopt even for medieval manuscripts.

The ruling classification which is – to our knowledge – new in the context of ruling analysis allows for rejecting *void* supporting material. Furthermore, unknowns such as the dominant ruling orientation, horizontal and/or vertical ruling lines, and the line spacing are known after classification which improve the results of the ruling line extraction. The polar power spectrum features that are extracted for classifying the supporting material are newly introduced based on scientific studies of existing texture features. In addition a publicly available database is created which allows for comparisons of other ruling analysis algorithms with the proposed one.

Word localization is carried out using Profile Boxes which are capable of approximating the word's *x-height*. The text classification is based on gradient features that have a compact representation. The evaluation shows that GSFs are capable of correctly representing text in the presence of noise and multiple authors or fonts. For text line clustering a methodology is introduced that minimizes the global error. Evaluations on the Handwriting Segmentation Contests show that this methodology competes with sate-of-the-art algorithms. Additional empirical evaluations on medieval manuscripts, modern printed pages, and real world Stasi data prove that the clustering is able to adopt to different document analysis scenarios.

The text localization and clustering adopt to local skew changes. However, if text which is perpendicular to the dominant orientation is present in a page, the text clustering fails. Hence, a clustering and word localization which is invariant with respect to local text orientation would improve the results for such scenarios.

Currently, text is classified into *noise*, *manuscript*, and *printed*. The empirical evaluations show that these classes can be adopted to other document analysis tasks. Nevertheless, a more sophisticated training could recognize the text analyzed at a finer granularity which would allow for distinguishing between different fonts or authors. This would contribute to the richness of document description.

The voting scheme currently utilizes knowledge about text composition in the sense that voting is performed on text lines and text blocks using a hierarchical structure. By these means a training stage is not needed for voting. If a document corpus had enough training samples, an undirected graph such as an MRF which is proposed by Zheng et al. [ZLD04] could improve the results.

To summarize, the system presented is fast at analyzing both, document snippets and whole pages. It extracts attributes that are especially designed for grouping and accurately aligning heterogeneous documents. At processing stages such as text localization or text line clustering a trade-off between computational complexity and accuracy is dispositive for the final algorithmic design.

“Das Rätsel ist gelöst,
die Forschung abgeschlossen.”

– Roland Jahn

List of acronyms

BB	Bounding Box
BoW	Bag of (Visual) Words
Cblv	Character Block Layout Variance
CC	Connected Component
Cch	Chain Code Histogram
Clpsd	Character Lowermost Point Standard Deviation
Crf	Conditional Random Field
Cubs	Center for Unified Biometrics and Sensors
Cvl-db	Computer Vision Lab Database [Kle+13]
Cvpr	Computer Vision and Patter Recognition
Das	Document Analysis Systems
DBscan	Density-Based Spatial Clustering of Applications with Noise
Dct	Discrete Cosine Transform
Dft	Discrete Fourier Transform
Dibco	Document Image Binarization Contest
Din	Deutsches Institut für Normung
DoG	Difference-of-Gaussians
dpi	Dots per Inch
DR	Detection Rate
Drf	Discriminative Random Field
Dscc	Directional Single-Connected Chains

ECM	Edge Co-occurrence Matrix
EDM	Entity Detection Metric
EM	Energy Minimization
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FM	Performance Metric
fn	False Negative
FNNC	Focus Nearest Neighbor Clustering
fp	False Positive
GMM	Gaussian Mixture Model
GRUHD	GReek Unconstrained HanDwriting Database [Kav+01]
GSF	Gradient Shape Feature
GVF	Gradient Vector Flow
HMM	Hidden Markov Model
IAM-DB	IAM Database [MB99]
ICDAR	International Conference on Document Analysis and Recognition
ICFHR	International Conference on Frontiers in Handwriting Recognition
IEEE	Institute of Electrical and Electronics Engineers
IJDAR	International Journal on Document Analysis and Recognition
IMPACT	Improving Access to Text
k-NN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
LPP	Local Projection Profile
MLP	Multi-Layer Perceptron
MRF	Markov Random Field
MST	Minimal Spanning Tree
NIST	National Institute of Standards and Technology

NP	Non-deterministic Polynomial-time
NN	Neural Network
NSM	Newspaper Segmentation Metric
OCR	Optical Character Recognition
PAGE	Page Analysis and Groundtruth Elements
PCA	Principal Component Analysis
PDF	Probability Density Function
PHR	Pixel-Level Hit Rate
PP	Projection Profile
PRImA	Pattern Recognition and Image Analysis Research Lab
RA	Recognition Accuracy
RBF	Radial Basis Function
RLS	Run Length Smoothing
ROC	Receiver Operating Characteristic
SC	Shape Context
SIFT	Scale Invariant Feature Transform
SDLFD	Spectrum-Domain Local Fluctuation Detection
SVM	Support Vector Machine
tp	True Positive
tn	True Negative
VC	Vapnik-Chervonenkis
XML	eXtended Markup Language

Bibliography

[ABG05] Apostolos Antonacopoulos, David Bridson, and Basilios Gatos. "Page Segmentation Competition". In: *Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), 29 August - 1 September 2005, Seoul, Korea*. IEEE Computer Society, 2005, pp. 75–79.

[ABR64] M. A. Aizerman, E. A. Braverman, and L. Rozonoer. "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning." In: *Automation and Remote Control,* vol. 25. 1964, pp. 821–837.

[AD09] Mudit Agrawal and David S. Doermann. "Voronoi++: A Dynamic Page Segmentation Approach Based on Voronoi and Docstrum Features". In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 1011–1015.

[AGB07] Apostolos Antonacopoulos, Basilios Gatos, and David Bridson. "Page Segmentation Competition". In: *9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September, Curitiba, Paraná, Brazil*. Ed. by Werner Bob. IEEE Computer Society, 2007, pp. 1279–1283.

[AGK03] Apostolos Antonacopoulos, Basilios Gatos, and Dimosthenis Karatzas. "ICDAR 2003 Page Segmentation Competition". In: *7th International Conference on Document Analysis and Recognition (ICDAR 2003), 2-Volume Set, 3-6 August 2003, Edinburgh, Scotland, UK*. IEEE Computer Society, 2003, pp. 688–692.

[AGP10] Marios Anthimopoulos, Basilios Gatos, and Ioannis Pratikakis. "A Two-Stage Scheme for Text Detection in Video Images". In: *Image Vision Computing* 28.9 (2010), pp. 1413–1426.

[Akb+10] S. Akbarpour, M. Sulaiman, N. Mustapha, and R.W. Rahmat. "Discriminating the Machine-Printed and Hand-Written Words Based on Legibility". In: *Seventh International Conference on Information Technology: New Generations (ITNG)*. 2010, pp. 364–369.

[AKD09] Wael Abd-Almageed, Jayant Kumar, and David S. Doermann. "Page Rule-Line Removal Using Linear Subspaces in Monochromatic Handwritten Arabic Documents". In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 768–772.

[Ant+09a] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos. "ICDAR 2009 Page Segmentation Competition". In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, July 2009, pp. 1370–1374.

[Ant+09b] Apostolos Antonacopoulos, David Bridson, Christos Papadopoulos, and Stefan Pletschacher. "A Realistic Dataset for Performance Evaluation of Document Layout Analysis". In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 296–300.

[Ant+11] Apostolos Antonacopoulos, Christian Clausner, Christos Papadopoulos, and Stefan Pletschacher. "Historical Document Layout Analysis Competition". In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 1516–1520.

[Ant+13] Apostolos Antonacopoulos, Christian Clausner, Christos Papadopoulos, and Stefan Pletschacher. "ICDAR 2013 Competition on Historical Newspaper Layout Analysis (HNLA 2013)". In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O'Conner. IEEE, 2013, pp. 1454–1458.

[AWH05] George B. Arfken, Hans J. Weber, and Frank E. Harris. *Mathematical Methods for Physicists, Sixth Edition: A Comprehensive Guide*. 6th ed. Academic Press, July 5, 2005.

[Bai94] Henry S. Baird. "Background Structure in Document Images". In: *International Journal of Pattern Recognition and Artificial Intelligence* 8.5 (1994), pp. 1013–1030.

[BC12] Purnendu Banerjee and B. B. Chaudhuri. "A System for Handwritten and Machine-Printed Text Separation in Bangla Document Images". In: *Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition*. 2012, pp. 758–762.

[BGV92] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the 20th Annual Conference on Learning Theory*. 1992, pp. 144–152.

[BI11] Micheal Baechler and Rolf Ingold. “Multi Resolution Layout Analysis of Medieval Manuscripts Using Dynamic MLP”. In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 1185–1189.

[BMP01] Serge Belongie, Jitendra Malik, and Jan Puzicha. “Matching Shapes”. In: *International Conference on Computer Vision*. 2001, pp. 454–463.

[BMP02] Serge Belongie, Jitendra Malik, and Jan Puzicha. “Shape Matching and Object Recognition Using Shape Contexts”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.4 (2002), pp. 509–522.

[Bob01] Werner Bob, ed. *6th International Conference on Document Analysis and Recognition (ICDAR 2001), 10-13 September 2001, Seattle, WA, USA*. IEEE Computer Society, 2001.

[Bob07] Werner Bob, ed. *9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September, Curitiba, Paraná, Brazil*. IEEE Computer Society, 2007.

[Bob99] Werner Bob, ed. *5th International Conference on Document Analysis and Recognition, (ICDAR 1999), 20-22 September, 1999, Bangalore, India*. IEEE Computer Society, 1999.

[Bor09] F. Bortolozzi, ed. *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. IEEE Computer Society, 2009.

[BPU12] Michael Blumenstein, Umapada Pal, and Seiichi Uchida, eds. *10th IAPR International Workshop on Document Analysis Systems, DAS 2012, Gold Coast, Queenslands, Australia, March 27-29, 2012*. IEEE, 2012.

[BSB08] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. “Segmentation of Curled Textlines Using Active Contours”. In: *The Eighth IAPR International Workshop on Document Analysis Systems, DAS 2008, September 16-19, 2008, Nara, Japan*. Ed. by Koichi Kise and Hiroshi Sako. IEEE Computer Society, 2008, pp. 270–277.

[BSB11a] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. “High Performance Layout Analysis of Arabic and Urdu Document Images”. In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 1275–1279.

[BSB11b] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. “Text-Line Extraction Using a Convolution of Isotropic Gaussian Filter with a Set of Line Filters”. In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 579–583.

[BSD13] Abdel Belaid, K. C. Santosh, and Vincent Poulain D'Andecy. "Handwritten and Printed Text Separation in Real Document". In: *Proceedings of IAPR Conference on Machine Vision Applications* abs/1303.4614 (2013).

[BSM99] Issam Bazzi, Richard M. Schwartz, and John Makhoul. "An Omnifont Open-Vocabulary OCR System for English and Arabic". In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 21.6 (1999), pp. 495–504.

[Can86] J Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 8.6 (June 1986), pp. 679–698.

[CFP10] Sukalpa Chanda, Katrin Franke, and Umapada Pal. "Document-Zone Classification in Torn Documents". In: *International Conference on Frontiers in Handwriting Recognition, ICFHR 2010, Kolkata, India, 16-18 November 2010*. Ed. by Patrick Kellenberger. IEEE Computer Society, 2010, pp. 25–30.

[Cho+07] S. P. Chowdhury, Sekhar Mandal, Amit K. Das, and Bhabatosh Chanda. "Segmentation of Text and Graphics from Document Images". In: *9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September, Curitiba, Paraná, Brazil*. Ed. by Werner Bob. IEEE Computer Society, 2007, pp. 619–623.

[CL14] Jin Chen and Daniel P. Lopresti. "Model-based ruling line detection in noisy handwritten documents". In: *Pattern Recognition Letters* Vol. 35 (2014), pp. 34–45.

[CP43] Warren Mc Culloch and Walter Pitts. "A Logical Calculus of Ideas Immanent in Nervous Activity". In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 127–147.

[CPA11a] Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. "Aletheia - An Advanced Document Layout and Text Ground-Truthing System for Production Environments". In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 48–52.

[CPA11b] Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. "Scenario Driven In-depth Performance Evaluation of Document Layout Analysis Methods". In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 1404–1408.

[Cro84] Franklin C. Crow. "Summed-area Tables for Texture Mapping". In: *SIGGRAPH Comput. Graph.* 18.3 (Jan. 1984), pp. 207–212.

[CV95] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pp. 273–297.

[CYL13] Kai Chen, Fei Yin, and Cheng-Lin Liu. “Hybrid Page Segmentation with Efficient Whitespace Rectangles Extraction and Grouping”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 958–962.

[Dav+96] Franck Davoine, Marc Antonini, Jean-Marc Chassery, and Michel Barlaud. “Fractal Image Compression Based on Delaunay Triangulation and Vector Quantization”. In: *IEEE Transactions on Image Processing* 5.2 (1996), pp. 338–346.

[Der13] Dmitry Deryagin. “Unified Performance Evaluation for OCR Zoning: Calculating Page Segmentation’s Score, That Includes Text Zones, Tables and Non-text Objects”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 953–957.

[Die+14] Markus Diem, Florian Kleber, Stefan Fiel, and Robert Sablatnig. “Semi-Automated Document Image Clustering and Retrieval”. In: *Document Recognition and Retrieval*. 2014.

[DKS09] Markus Diem, Florian Kleber, and Robert Sablatnig. “Analysis of Document Snippets as a Basis for Reconstruction”. In: *Proceedings of the 10th International Symposium on Virtual Reality, Archaeology, and Cultural Heritage*. Ed. by Kurt Debattista, Cinzia Perlingieri, Denis Pitzalis, and Sandro Spina. 2009, pp. 101–108.

[DKS10] Markus Diem, Florian Kleber, and Robert Sablatnig. “Document Analysis Applied to Fragments: Feature Set for the Reconstruction of Torn Documents”. In: *Proceedings of the 9th International Workshop on Document Analysis Systems*. Ed. by D. Doermann, V. Govindaraju, D. Lopresti, and P. Natarajan. Boston, USA, June 2010, pp. 393–400.

[DKS11] Markus Diem, Florian Kleber, and Robert Sablatnig. “Text Classification and Document Layout Analysis of Torn Documents”. In: *Proceedings of the 11th International Conference on Document Analysis and Reconstruction (ICDAR 2011)*. Beijing, China: IEEE, 2011, pp. 1181–1184.

[DKS12] Markus Diem, Florian Kleber, and Robert Sablatnig. “Skew Estimation of Sparsely Inscribed Document Fragments”. In: *10th IAPR International Workshop on Document Analysis Systems, DAS 2012, Gold Coast, Queenslands, Australia, March 27-29, 2012*. Ed. by Michael Blumenstein, Umapada Pal, and Seiichi Uchida. IEEE, 2012, pp. 292–296.

[DKS13] Markus Diem, Florian Kleber, and Robert Sablatnig. “Text Line Detection for Heterogeneous Documents”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 743–747.

[DL01] D. Doermann and J. Liang. “Binary Document Image Using Similarity Multiple Texture Features”. In: *Proceedings of Symposium on Document Image Understanding Technology*. 2001, pp. 181–193.

[DPB09] Xiaojun Du, Wumo Pan, and Tien D. Bui. “Text line segmentation in handwritten documents using Mumford-Shah model”. In: *Pattern Recognition* 42.12 (2009), pp. 3136–3145.

[DS10] Markus Diem and Robert Sablatnig. “Are Characters Objects?” In: *Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Kolkata, India, 2010, pp. 565–570.

[EDC97] Kamran Etemad, David S. Doermann, and Rama Chellappa. “Multiscale Segmentation of Unstructured Document Pages Using Soft Decision Integration”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 19.1 (1997), pp. 92–96.

[Esp+90] F. Esposito, D. Malerba, G. Semeraro, E. Annese, and G. Scafuro. “An Experimental Page Layout Recognition System for Office Document Automatic Classification: An Integrated Approach for Inductive Generalization”. In: *Proceedings of the 10th International Conference on Pattern Recognition, 1990.* Vol. 1. 1990, pp. 557–562.

[FHD90] J.L. Fisher, S.C. Hinds, and D.P. D’Amato. “A Rule-Based System for Document Image Segmentation”. In: *Proceedings of the 10th International Conference on Pattern Recognition, 1990.* Vol. 1. 1990, pp. 567–572.

[Fis+10] Andreas Fischer, Emanuel Indermühle, Horst Bunke, Gabriel Viehhauser, and Michael Stolz. “Ground Truth Creation for Handwriting Recognition in Historical Documents”. In: *The Ninth IAPR International Workshop on Document Analysis Systems, DAS 2010, June 9-11, 2010, Boston, Massachusetts, USA*. Ed. by David S. Doermann, Venu Govindaraju, Daniel P. Lopresti, and Premkumar Natarajan. ACM International Conference Proceeding Series. ACM, 2010, pp. 3–10.

[Fis+11] Andreas Fischer, Emanuel Indermühle, Volkmar Frinken, and Horst Bunke. “HMM-Based Alignment of Inaccurate Transcriptions for Historical Documents”. In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 53–57.

[Fis+12] Andreas Fischer, Andreas Keller, Volkmar Frinken, and Horst Bunke. “Lexicon-free handwritten word spotting using character HMMs”. In: *Pattern Recognition Letters* 33.7 (2012), pp. 934–942.

[Fis36] R. A. Fisher. “The Use of Multiple Measurements in Taxonomic Problems”. In: *Annals of Eugenics* 7.7 (1936), pp. 179–188.

[FK88] L.A. Fletcher and R. Kasturi. “A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.6 (1988), pp. 910–918.

[Flu00] Jan Flusser. “On the Independence of Rotation Moment Invariants”. In: *Pattern Recognition* 33.9 (2000), pp. 1405–1410.

[Flu02] Jan Flusser. “On the Inverse Problem of Rotation Moment Invariants”. In: *Pattern Recognition* 35.12 (2002), pp. 3015–3017.

[Flu05] Jan Flusser. *Moment Invariants in Image Analysis*. `http://559t.iki.rssi.ru/~vgrishin/Course/Jan_Flusser,_Barbara_Zitova,_Tomas_Suk-Moments_and_Moment_Invariants_in_Pattern_Recognition-Wiley(2009).pdf`. [Online, accessed 02-March-2014]. 2005.

[Fri+12] Volkmar Frinken, Andreas Fischer, R. Manmatha, and Horst Bunke. “A Novel Word Spotting Method Based on Recurrent Neural Networks”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 34.2 (2012), pp. 211–224.

[FS13] S Fiel and R Sablatnig. “Writer Identification and Writer Retrieval using the Fisher Vector on Visual Vocabularies”. In: *2013 International Conference on Document Analysis and Recognition*. 2013, pp. 545–549.

[FS93] Jan Flusser and Tomás Suk. “Pattern recognition by affine moment invariants”. In: *Pattern Recognition* 26.1 (1993), pp. 167–174.

[FSG06] Faisal Farooq, Karthik Sridharan, and Venu Govindaraju. “Identifying Handwritten Text in Mixed Documents”. In: *18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China*. Ed. by Y.Y. Tang, S.P. Wang, G. Lorette, D.S. Yeung, and H. Yan. IEEE Computer Society, 2006, pp. 1142–1145.

[FWT98] Kuo-Chin Fan, Liang-Shen Wang, and Yin-Tien Tu. “Classification Of Machine-Printed And Handwritten Texts Using Character Block Layout Variance”. In: *Pattern Recognition* 31.9 (1998), pp. 1275–1284.

[Gar+12] Angelika Garz, Andreas Fischer, Robert Sablatnig, and Horst Bunke. “Binarization-Free Text Line Segmentation for Historical Documents Based on Interest Point Clustering”. In: *10th IAPR International Workshop on Document Analysis Systems, DAS 2012, Gold Coast, Queenslands, Australia, March 27-29, 2012*. Ed. by Michael Blumenstein, Umapada Pal, and Seiichi Uchida. IEEE, 2012, pp. 95–99.

[Gar+13] Angelika Garz, Andreas Fischer, Horst Bunke, and Rolf Ingold. “A Binarization-Free Clustering Approach to Segment Curved Text Lines in Historical Manuscripts”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 1290–1294.

[GAS07] Basilios Gatos, Apostolos Antonacopoulos, and Nikolaos Stamatopoulos. “Handwriting Segmentation Contest”. In: *9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September, Curitiba, Paraná, Brazil*. Ed. by Werner Bob. IEEE Computer Society, 2007, pp. 1284–1288.

[GM01] J.K. Guo and M.Y. Ma. “Separating Handwritten Material from Machine Printed Text using Hidden Markov Models”. In: *Proceedings of the Sixth International Conference on Document Analysis and Recognition*. 2001, pp. 439–443.

[GMA01] B. Gatos, S. L. Mantzaris, and A. Antonacopoulos. “First International Newspaper Segmentation Contest”. In: *6th International Conference on Document Analysis and Recognition (ICDAR 2001), 10-13 September 2001, Seattle, WA, USA*. Ed. by Werner Bob. IEEE Computer Society, 2001, pp. 1190–.

[GPV13] Albert Gordo, Florent Perronnin, and Ernest Valveny. “Large-scale document image retrieval and classification with runlength histograms and binary embeddings”. In: *Pattern Recognition* 46.7 (2013), pp. 1898–1905.

[GSD11] Angelika Garz, Robert Sablatnig, and Markus Diem. “Layout Analysis for Historical Manuscripts Using Sift Features”. In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 508–512.

[GSL09] Basilios Gatos, Nikolaos Stamatopoulos, and Georgios Louloudis. “ICDAR 2009 Handwriting Segmentation Contest”. In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 1393–1397.

[GSL10] Basilios Gatos, Nikolaos Stamatopoulos, and Georgios Louloudis. “ICFHR 2010 Handwriting Segmentation Contest”. In: *International Conference on Frontiers in Handwriting Recognition, ICFHR 2010, Kolkata, India, 16-18 November 2010*. Ed. by Patrick Kellenberger. IEEE Computer Society, 2010, pp. 737–742.

[Han+13] M. Hangarge, K. C. Santosh, S. Doddamani, and R. Pardeshi. “Statistical Texture Features based Handwritten and Printed Text Classification in South Indian Documents”. In: *Proceedings of International Conference on Emerging Trends in Electrical, Communications, and Information Technologies*. 2013, pp. 215–221.

[HE03] Greg Hamerly and Charles Elkan. “Learning the k in k-means”. In: *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*. Ed. by Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf. MIT Press, 2003.

[HL93] F. Hönes and J. Lichter. “Text String Extraction within Mixed-Mode Documents”. In: *Proceedings of the Second International Conference on Document Analysis and Recognition*. Oct. 1993, pp. 655–659.

[HS88] Chris Harris and Mike Stephens. “A Combined Corner and Edge Detector”. In: *Proceedings of the Alvey Vision Conference, AVC 1988, Manchester, UK, September, 1988*. Ed. by Christopher J. Taylor. Alvey Vision Club, 1988, pp. 1–6.

[HSD73] R.M. Haralick, K. Shanmugam, and Its'Hak Dinstein. “Textural Features for Image Classification”. In: *IEEE Transactions on Systems, Man and Cybernetics* SMC-3.6 (1973), pp. 610–621.

[Hu62] Ming-Kuei Hu. “Visual pattern recognition by moment invariants”. In: *IRE Transactions on Information Theory* 8.2 (1962), pp. 179–187.

[IB93] D.J. Ittner and H.S. Baird. “Language-Free Layout Analysis”. In: *Proceedings of the Second International Conference on Document Analysis and Recognition*. Oct. 1993, pp. 336–340.

[ITW93] S. Imade, S. Tatsuta, and T. Wada. “Segmentation and Classification for Mixed Text/Image Documents using Neural Network”. In: *Proceedings of the Second International Conference on Document Analysis and Recognition*. 1993, pp. 930–934.

[JBW99] Xiaoyi Jiang, H. Bunke, and D. Widmer-Kljajo. “Skew Detection of Document Images by Focused Nearest-Neighbor Clustering”. In: *Proceedings of the Fifth International Conference on Document Analysis and Recognition, 1999. ICDAR '99.* Sept. 1999, pp. 629–632.

[JJN04] Seung Ick Jang, Seon-Hwa Jeong, and Yun-Seok Nam. “Classification of Machine-Printed and Handwritten Addresses on Korean Mail Piece Images using Geometric Features”. In: *Proceedings of the 17th International Conference on Pattern Recognition*. Vol. 2. 2004, 383–386 Vol.2.

[JSS99] Min-Chul Jung, Y.-C. Shin, and S.N. Srihari. “Multifont Classification using Typographical Attributes”. In: *Proceedings of the Fifth International Conference on Document Analysis and Recognition*. 1999, pp. 353–356.

[JZ96] Anil K. Jain and Yu Zhong. “Page Segmentation using Texture Analysis”. In: *Pattern Recognition* 29.5 (1996), pp. 743–770.

[Kan+07] R. Kandan, Nirup Kumar Reddy, K. R. Arvind, and A. G. Ramakrishnan. “A Robust Two Level Classification Algorithm for Text Localization in Documents”. In: *Advances in Visual Computing, Third International Symposium, ISVC 2007, Lake Tahoe, NV, USA, November 26-28, 2007, Proceedings, Part II*. Ed. by George Bebis, Richard D. Boyle, Bahram Parvin, Darko Koracin, Nikos Paragios, Tanveer Fathima Syeda-Mahmood, Tao Ju, Zicheng Liu, Sabine Coquillart, Carolina Cruz-Neira, Torsten Müller, and Thomas Malzbender. Vol. 4842. Lecture Notes in Computer Science. Springer, 2007, pp. 96–105.

[Kav+01] Ergina Kavallieratou, Nikos Liolios, E. Koutsogeorgos, Nikos Fakotakis, and George K. Kokkinakis. “The GRUHD Database of Greek Unconstrained Handwriting”. In: *6th International Conference on Document Analysis and Recognition (ICDAR 2001), 10-13 September 2001, Seattle, WA, USA*. Ed. by Werner Bob. IEEE Computer Society, 2001, pp. 561–565.

[KC10] Hyung il Koo and Ik Cho Nam. “State Estimation in a Document Image and Its Application in Text Block Identification and Text Line Extraction”. In: *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Vol. 6312. Lecture Notes in Computer Science. Springer, 2010, pp. 421–434.

[KC12] Hyung il Koo and Ik Cho Nam. “Text-Line Extraction in Handwritten Chinese Documents Based on an Energy Minimization Framework”. In: *IEEE Transactions on Image Processing* 21.3 (2012), pp. 1169–1175.

[KDS09] Florian Kleber, Markus Diem, and Robert Sablatnig. “Torn Document Analysis as a Prerequisite for Reconstruction”. In: *Proceedings of the 15th International Conference on Virtual Systems and MultiMedia (VSMM 2009)*. 2009, pp. 143–148.

[KDS11] Florian Kleber, Markus Diem, and Robert Sablatnig. “Scale Space Binarization Using Edge Information Weighted by a Foreground Estimation”. In: *Proceedings of the 11th International Conference on Document Analysis and Reconstruction (ICDAR 2011)*. Beijing, China: IEEE, 2011, pp. 854–858.

[Kel10] Patrick Kellenberger, ed. *International Conference on Frontiers in Handwriting Recognition, ICFHR 2010, Kolkata, India, 16-18 November 2010*. IEEE Computer Society, 2010.

[Kel11] P. Kellenberger, ed. *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. IEEE, 2011.

[KH06] Sanjiv Kumar and Martial Hebert. “Discriminative Random Fields”. In: *International Journal of Computer Vision* 68.2 (2006), pp. 179–201.

[KH12] Yuuya Konno and Akira Hirose. “Early-Vision-Inspired Method to Distinguish between Handwritten and Machine-Printed Character Images Using Hough Transform”. In: *Neural Information Processing - 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part V*. Ed. by Tingwen Huang, Zhigang Zeng, Chuandong Li, and Chi-Sing Leung. Vol. 7667. Lecture Notes in Computer Science. Springer, 2012, pp. 353–360.

[KH90] A. Khotanzad and Yaw Hua Hong. “Invariant Image Recognition by Zernike Moments”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.5 (1990), pp. 489–497.

[KHK08] Jumpei Koyama, Akira Hirose, and Masahiro Kato. “Local-Spectrum-Based Distinction between Handwritten and Machine-Printed Characters”. In: *Proceedings of the International Conference on Image Processing, ICIP 2008, October 12-15, 2008, San Diego, California, USA*. IEEE, 2008, 1021–1024.

[KKH08] Jumpei Koyama, Masahiro Kato, and Akira Hirose. “Distinction between Handwritten and Machine-Printed Characters with no Need to Locate Character or Text Line Position”. In: *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*. IEEE, 2008, pp. 4044–4051.

[Kle+11] Florian Kleber, Markus Diem, Fabian Hollaus, Martin Lettner, Robert Sablatnig, Melanie Gau, and Heinz Miklas. “Technical Approaches to Manuscript Analysis and Reconstruction”. In: *New Approaches to Book and Paper Conservation-Restoration*. Ed. by Patricia Engel, Jospeh Schiro, Rene Larsen, Elissaveta Moussakova, and Istvan Kecskemeti. Horn, Austria: Verlag Berger, 2011, pp. 533–558.

[Kle+13] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. “CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 560–564.

[KPD92] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. “Handwritten Digit Recognition by Neural Networks with Single-Layer Training”. In: *IEEE Transactions on Neural Networks* 3.6 (1992), pp. 962–968.

[KSA04] Ergina Kavallieratou, Efstathios Stamatatos, and Hera Antonopoulou. “Machine-Printed from Handwritten Text Discrimination”. In: *Ninth International Workshop on Frontiers in Handwriting Recognition, IWFHR-9 2004, Kokubunji, Tokyo, Japan, October 26-29, 2004*. IEEE Computer Society, 2004, pp. 312–316.

[KSI98] Koichi Kise, Akinori Sato, and Motoi Iwata. “Segmentation of Page Images Using the Area Voronoi Diagram”. In: *Computer Vision and Image Understanding* 70.3 (1998), pp. 370–382.

[KSK95] K. Kuhnke, L. Simoncini, and Z. M. Kovács-Vajna. “A System for Machine-Written and Hand-Written Character Distinction”. In: *ICDAR*. 1995, pp. 811–814.

[Kum+11] Jayant Kumar, Rohit Prasad, Huaigu Cao, Wael Abd-Almageed, David S. Doermann, and Premkumar Natarajanku. “Shape Codebook based Handwritten and Machine Printed Text Zone Extraction”. In: *Document Recognition and Retrieval XVIII - DRR 2011, 18th Document Recognition and*

Retrieval Conference, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 24-29, 2011, Proceedings. Ed. by Gady Agam and Christian Viard-Gaudin. Vol. 7874. SPIE Proceedings. SPIE, 2011, pp. 1–10.

[Li+08] Yi Li, Yefeng Zheng, David Doermann, and Stefan Jaeger. "Script-Independent Text Line Segmentation in Freestyle Handwritten Documents". In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 30.8 (2008), pp. 1313–1329.

[Liu+01] Fei Liu, Yupin Luo, Dongcheng Hu, and Masataka Yoshikawa. "A New Component Based Algorithm for Newspaper Layout Analysis". In: *6th International Conference on Document Analysis and Recognition (ICDAR 2001), 10-13 September 2001, Seattle, WA, USA*. Ed. by Werner Bob. IEEE Computer Society, 2001, pp. 1176–1180.

[LK10] Daniel P. Lopresti and Ergina Kavallieratou. "Ruling Line Removal in Handwritten Page Images". In: *20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*. Ed. by Juen Guerrero. IEEE, 2010, pp. 2704–2707.

[Lou+09] Georgios Louloudis, Basilios Gatos, Ioannis Pratikakis, and Constantin Halatsis. "Text Line and Word Segmentation of Handwritten Documents". In: *Pattern Recognition* 42.12 (2009), pp. 3169–3183.

[Low04] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[Low99] David G. Lowe. "Object Recognition from Local Scale-Invariant Features". In: *Proceedings of the International Conference on Computer Vision*. 1999, pp. 1150–1157.

[LP05] Fei-Fei Li and Pietro Perona. "A Bayesian Hierarchical Model for Learning Natural Scene Categories". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. IEEE Computer Society, 2005, pp. 524–531.

[LZT07] Laurence Likforman-Sulem, Abderrazak Zahour, and Bruno Taconet. "Text Line Segmentation of Historical Documents: A Survey". In: *International Journal on Document Analysis and Recognition* 9.2-4 (2007), pp. 123–138.

[MB02] U.-V. Marti and H. Bunke. "The IAM-database: An English Sentence Database for Offline Handwriting Recognition". In: *International Journal on Document Analysis and Recognition* 5.1 (1 2002), pp. 39–46.

[MB12] Saeed Mozaffari and Parnia Bahar. "Farsi/Arabic Handwritten from Machine-Printed Words Discrimination". In: *Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition*. 2012, pp. 698–703.

[MB99] Urs-Viktor Marti and Horst Bunke. “A Full English Sentence Database for Off-line Handwriting Recognition”. In: *5th International Conference on Document Analysis and Recognition, (ICDAR 1999), 20-22 September, 1999, Bangalore, India*. Ed. by Werner Bob. IEEE Computer Society, 1999, pp. 705–708.

[MK01] Song Mao and Tapas Kanungo. “Empirical Performance Evaluation Methodology and its Application to Page Segmentation Algorithms”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.3 (2001), pp. 242–256.

[Mon+09] Florent Montreuil, Emmanuele Grosicki, Laurent Heutte, and Stéphane Nicolas. “Unconstrained Handwritten Document Layout Extraction Using 2D Conditional Random Fields”. In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 853–857.

[Mon+10] Florent Montreuil, Stéphane Nicolas, Emmanuele Grosicki, and Laurent Heutte. “A New Hierarchical Handwritten Document Layout Extraction Based on Conditional Random Field Modeling”. In: *International Conference on Frontiers in Handwriting Recognition, ICFHR 2010, Kolkata, India, 16-18 November 2010*. Ed. by Patrick Kellenberger. IEEE Computer Society, 2010, pp. 31–36.

[MS05] Krystian Mikolajczyk and Cordelia Schmid. “A Performance Evaluation of Local Descriptors”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.10 (2005), pp. 1615–1630.

[NG12] S. Narayan and S.D. Gowda. “Discrimination of Handwritten and Machine Printed Text in Scanned Document Images based on Rough Set Theory”. In: *World Congress on Information and Communication Technologies (WICT)*. 2012, pp. 590–594.

[Nik+10] Nikos A. Nikolaou, Michael Makridis, Basilios Gatos, Nikolaos Stamatopoulos, and Nikos Papamarkos. “Segmentation of Historical Machine-Printed Documents using Adaptive Run Length Smoothing and Skeleton Segmentation Paths”. In: *Image Vision Computing* 28.4 (2010), pp. 590–604.

[NS84] G. Nagy and S. Seth. “Hierarchical Representation of Optically Scanned Documents”. In: *Proceedings of the 7th International Conference on Pattern Recognition*. Montreal, Canada, 1984, pp. 347–349.

[OCo13] Lisa O’Conner, ed. *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. IEEE, 2013.

[OGo93] Lawrence O’Gorman. “The Document Spectrum for Page Layout Analysis”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 15.11 (1993), pp. 1162–1173.

[OT93] M. Okamoto and M. Takahashi. “A Hybrid Page Segmentation Method”. In: *Proceedings of the Second International Conference on Document Analysis and Recognition*. Oct. 1993, pp. 743–746.

[Ots79] Nobuyuki Otsu. “A Threshold Selection Method from Gray-Level Histograms”. In: *IEEE Transactions on Systems, Man and Cybernetics* 9.1 (1979), pp. 62–66.

[Pal+07] Umapada Pal, Nabin Sharma, Tetsushi Wakabayashi, and Fumitaka Kimura. “Handwritten Numeral Recognition of Six Popular Indian Scripts”. In: *9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September, Curitiba, Paraná, Brazil*. Ed. by Werner Bob. IEEE Computer Society, 2007, pp. 749–753.

[Pap+10] Vassilios Papavassiliou, Themos Stafylakis, Vassilios Katsouros, and George Carayannis. “Handwritten Document Image Segmentation Into Text Lines and Words”. In: *Pattern Recognition* 43.1 (2010), pp. 369–377.

[Pap+13] A. Papandreou, Basilios Gatos, Georgios Louloudis, and Nikolaos Stamatopoulos. “ICDAR 2013 Document Image Skew Estimation Contest (DISEC 2013)”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 1444–1448.

[PB11] Samuel J. Pinson and William A. Barrett. “Connected Component Level Discrimination of Handwritten and Machine-Printed Text Using Eigenfaces”. In: *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*. Ed. by P. Kellenberger. IEEE, 2011, pp. 1394–1398.

[PC01] U Pal and BB Chaudhuri. “Machine-Printed and Hand-Written Text Lines Identification”. In: *Pattern Recognition Letters* 22.3-4 (2001), pp. 431–441.

[PC99] U. Pal and B. B. Chaudhuri. “Automatic Separation of Machine-Printed and Hand-Written Text Lines”. In: *5th International Conference on Document Analysis and Recognition, (ICDAR 1999), 20-22 September, 1999, Bangalore, India*. Ed. by Werner Bob. IEEE Computer Society, 1999, 645–648.

[Pen+09] Xujun Peng, Srirangaraj Setlur, Venu Govindaraju, Ramachandrula Sitaram, and Kiran Bhuvanagiri. “Markov Random Field Based Text Identification from Annotated Machine Printed Documents”. In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 431–435.

[Pen+11] Xujun Peng, Srirangaraj Setlur, Venu Govindaraju, and Ramachandrula Sitaram. “Handwritten Text Separation from Annotated Machine Printed Documents using Markov Random Fields”. In: *International Journal on Document Analysis and Recognition* 16.1 (2011), pp. 1–16.

[PGN10] Ioannis Pratikakis, Basilios Gatos, and Konstantinos Ntirogiannis. “H-DIBCO 2010 - Handwritten Document Image Binarization Competition”. In: *International Conference on Frontiers in Handwriting Recognition, ICFHR 2010, Kolkata, India, 16-18 November 2010*. Ed. by Patrick Kellenberger. IEEE Computer Society, 2010, pp. 727–732.

[PGN11] I. Pratikakis, B. Gatos, and K. Ntirogiannis. “ICDAR 2011 Document Image Binarization Contest (DIBCO 2011)”. In: *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR)*. 2011, pp. 1506–1510.

[PGN12] Ioannis Pratikakis, Basilios Gatos, and Konstantinos Ntirogiannis. “ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012)”. In: *Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition*. 2012, pp. 817–822.

[PGN13] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. “ICDAR 2013 Document Image Binarization Contest (DIBCO 2013)”. In: *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*. 2013, pp. 1471–1476.

[PZ91] T. Pavlidis and J Zhou. “Page Segmentation by White Stream”. In: *First International Conference on Document Analysis and Recognition*. St. Malo, France, 1991, pp. 945–953.

[Rah+06] Esa Rahtu, Mikko Salo, Janne Heikkilä, and Jan Flusser. “Generalized Affine Moment Invariants for Object Recogntion”. In: *18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China*. Ed. by Y.Y. Tang, S.P. Wang, G. Lorette, D.S. Yeung, and H. Yan. IEEE Computer Society, 2006, pp. 634–637.

[RC13] Ana Rebelo and Jaime S. Cardoso. “Staff Line Detection and Removal in the Grayscale Domain”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 57–61.

[RHW88] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning Internal Representations by Error Propagation”. In: *Neurocomputing: foundations of research*. Ed. by James A. Anderson and Edward Rosenfeld. Cambridge, MA, USA: MIT Press, 1988, pp. 673–695.

[SEB13] A. Saidani, A.Kacem Echi, and A. Belaid. “Identification of Machine-Printed and Handwritten Words in Arabic and Latin Scripts”. In: *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*. 2013, pp. 798–802.

[SK87] L. Sirovich and M. Kirby. “Low-Dimensional Procedure for the Characterization of Human Faces”. In: *Journal of Optical Society of America* 4.3 (Mar. 1, 1987), pp. 519–524.

[SKB08] Faisal Shafait, Daniel Keysers, and Thomas M. Breuel. “Performance Evaluation and Benchmarking of Six-Page Segmentation Algorithms”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 30.6 (2008), pp. 941–954.

[SL13] Feng Su and Tong Lu. “Discriminative Weighting and Subspace Learning for Ensemble Symbol Recognition”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 1088–1092.

[SLT10] Bolan Su, Shijian Lu, and Chew Lim Tan. “Binarization of Historical Document Images Using the Local Maximum and Minimum”. In: *DAS ’10: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. Boston, Massachusetts: ACM, 2010, pp. 159–166.

[Smi09] Raymond W. Smith. “Hybrid Page Layout Analysis via Tab-Stop Detection”. In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 241–245.

[SN08] Jan Schneider and Bertram Nickolay. “The Stasi puzzle”. In: *Fraunhofer Magazine, Special Issue* 1 (2008). Ed. by Beate Koch, pp. 32–33.

[Sof97] Aya Soffer. “Image Categorization Using Texture Features”. In: *ICDAR ’97: Proceedings of the 4th International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 233–237.

[SPJ97] Anikó Simon, Jean-Christophe Pret, and Peter Johnson. “A Fast Algorithm for Bottom-Up Document Layout Analysis”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 19.3 (1997), pp. 273–277.

[SSG09] Zhixin Shi, Srirangaraj Setlur, and Venu Govindaraju. “A Steerable Directional Local Profile Technique for Extraction of Handwritten Arabic Text Lines”. In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 176–180.

[Sta+08] Themos Stafylakis, Vassilios Papavassiliou, Vassilios Katsouros, and George Carayannis. “Robust Text-Line and Word Segmentation for Handwritten Documents Images”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*. IEEE, 2008, pp. 3393–3396.

[Sta+13] Nikolaos Stamatopoulos, Basilis Gatos, Georgios Louloudis, Umapada Pal, and Alireza Alaei. “ICDAR 2013 Handwriting Segmentation Contest”. In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O’Conner. IEEE, 2013, pp. 1402–1406.

[Tan+06] Y.Y. Tang, S.P. Wang, G. Lorette, D.S. Yeung, and H. Yan, eds. *18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China*. IEEE Computer Society, 2006.

[Tou83] Godfried Toussaint. "Solving Geometric Problems with the Rotating Calipers". In: *In Proceedings IEEE MELECON*. 1983, pp. 10–17.

[TP91] M.A. Turk and A.P. Pentland. "Face Recognition using Eigenfaces". In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*. 1991, pp. 586–591.

[Vap06] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data*. New York, USA: Springer Science and Business Media, Inc., 2006.

[Vap82] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1982.

[VC74] Vladimir Vapnik and Alexey Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974.

[VJ04] Paul A. Viola and Michael J. Jones. "Robust Real-Time Face Detection". In: *International Journal of Computer Vision* 57.2 (2004), pp. 137–154.

[Wam+12] Yannick Wamain, Jessica Tallet, Pier-Giorgio Zanone, and Marieke Longcamp. "Brain Responses to Handwritten and Printed Letters Differentially Depend on the Activation State of the Primary Motor Cortex". In: *NeuroImage* 63.3 (2012), pp. 1766–1773.

[WCW82] Kwan Y. Wong, Richard G. Casey, and Friedrich M. Wahl. "Document Analysis System". In: *IBM Journal of Research and Development* 26.6 (1982), pp. 647–656.

[Wei+13] Hao Wei, Micheal Baechler, Fouad Slimane, and Rolf Ingold. "Evaluation of SVM, MLP and GMM Classifiers for Layout Analysis of Historical Documents". In: *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*. Ed. by Lisa O'Conner. IEEE, 2013, pp. 1220–1224.

[WK77] Roy E. Welsch and Edwin Kuh. *Linear Regression Diagnostics*. Tech. rep. 923-77. Massachusetts Institute of Technology, Apr. 1977.

[WLW04] Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng. "Probability Estimates for Multi-class Classification by Pairwise Coupling". In: *Journal of Machine Learning Research* 5 (2004), pp. 975–1005.

[WP93] J. Wieser and A. Pinz. "Layout and Analysis: Finding Text, Titles, and Photos in Digital Images of Newspaper Pages". In: *Proceedings of the Second International Conference on Document Analysis and Recognition*. Oct. 1993, pp. 774–777.

[XY03] Yi Xiao and Hong Yan. “Text Region Extraction in a Document Image based on the Delaunay Tessellation”. In: *Pattern Recognition* 36.3 (2003), pp. 799–809.

[YL09] Fei Yin and Cheng-Lin Liu. “Handwritten Chinese Text Line Segmentation by Clustering with Distance Metric Learning”. In: *Pattern Recognition* 42.12 (2009), pp. 3146–3157.

[Yos+09] Itay Bar Yosef, Nate Hagbi, Klara Kedem, and Its’hak Dinstein. “Line Segmentation for Degraded Handwritten Historical Documents”. In: *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*. Ed. by F. Bortolozzi. IEEE Computer Society, 2009, pp. 1161–1165.

[YT11] Chucai Yi and YingLi Tian. “Text String Detection From Natural Scenes by Structure-Based Partition and Grouping”. In: *IEEE Transactions on Image Processing* 20.9 (2011), pp. 2594–2605.

[YW13] Hong Yan and Toyohide Watanabe. “Document Page Retrieval based on Geometric Layout Features”. In: *The 7th International Conference on Ubiquitous Information Management and Communication, ICUIMC ’13, Kota Kinabalu, Malaysia - January 17 - 19, 2013*. ACM, 2013, p. 60.

[YZD06] Li Yi, Yefeng Zheng, and David Doermann. “Detecting Text Line in Handwritten Documents”. In: *International Conference on Pattern Recognition, ICPR*. Hong Kong, China, 2006, pp. 1030–1033.

[Zag+12] Konstantinos Zagoris, Ioannis Pratikakis, Apostolos Antonacopoulos, Basilios Gatos, and Nikos Papamarkos. “Handwritten and Machine Printed Text Separation in Document Images Using the Bag of Visual Words Paradigm”. In: *ICFHR*. 2012, pp. 103–108.

[ZC11] Et-Tahir Zemouri and Youcef Chibani. “Machine Printed Handwritten Text Discrimination using Radon Transform and SVM Classifier”. In: *11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Córdoba, Spain, November 22-24, 2011*. Ed. by Sebastián Ventura, Ajith Abraham, Krzysztof J. Cios, Cristóbal Romero, Francesco Marcelloni, José Manuel Benítez, and Eva Lucrecia Gibaja Galindo. IEEE, 2011, pp. 1306–1310.

[ZF10] Majid Ziaratban and Karim Faez. “An Adaptive Script-Independent Block-Based Text Line Extraction”. In: *20th International Conference on Pattern Recognition (ICPR)*. 2010, pp. 249–252.

[ZF11] Majid Ziaratban and Karim Faez. “Adaptive Script-Independent Text Line Extraction”. In: *IEICE Transactions* 94-D.4 (2011), pp. 866–877.

[Zha+13] Luming Zhang, Mingli Song, Zicheng Liu, Xiao Liu, Jiajun Bu, and Chun Chen. “Probabilistic Graphlet Cut: Exploiting Spatial Structure Cue for Weakly Supervised Image Segmentation”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*. IEEE, 2013, pp. 1908–1915.

[Zhe+01] Yefeng Zheng, Changsong Liu, Xiaoqing Ding, and Shiyan Pan. “Form Frame Line Detection with Directional Single-Connected Chain”. In: *6th International Conference on Document Analysis and Recognition (ICDAR 2001), 10-13 September 2001, Seattle, WA, USA*. Ed. by Werner Bob. IEEE Computer Society, 2001, pp. 699–703.

[Zhu+09] Guangyu Zhu, Yefeng Zheng, David S. Doermann, and Stefan Jaeger. “Signature Detection and Matching for Document Image Retrieval”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 31.11 (2009), 2015–2031.

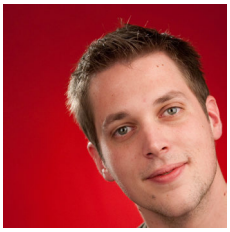
[ZL12] Xiaofeng Zhang and Yue Lu. “Handwritten and Machine Printed Text Discrimination using an Edge Co-Occurrence Matrix”. In: *Proceedings of the International Conference on Audio, Language and Image Processing (ICALIP)*. 2012, pp. 828–831.

[ZLD01] Yefeng Zheng, Changsong Liu, and Xiaoqing Ding. “Single-Character Type Identification”. In: *Electronic Imaging 2002*. International Society for Optics and Photonics. 2001, pp. 49–56.

[ZLD02] Yefeng Zheng, Huiping Li, and David S. Doermann. “The Segmentation and Identification of Handwriting in Noisy Document Images”. In: *Document Analysis Systems V, 5th International Workshop, DAS 2002, Princeton, NJ, USA, August 19-21, 2002, Proceedings*. Ed. by Daniel P. Lopresti, Jianying Hu, and Ramanujan S. Kashi. Vol. 2423. Lecture Notes in Computer Science. Springer, 2002, pp. 95–105.

[ZLD04] Yefeng Zheng, Huiping Li, and David S. Doermann. “Machine Printed Text and Handwriting Identification in Noisy Document Images”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 26.3 (2004), pp. 337–353.

[ZLD05] Yefeng Zheng, Huiping Li, and David S. Doermann. “A Parallel-Line Detection Algorithm Based on HMM Decoding”. In: *IEEE Transactions on Pattern Analysis Machine Intelligence* 27.5 (2005), pp. 777–792.



Markus Diem

since 2010	Senior Researcher for the *Document Information Retrieval (DIR)* project at the Computer Vision Lab, Vienna University of Technology
2007-10	MSc. Vienna University of Technology, Visual Computing
2002-07	BSc. Vienna University of Technology, Media and Computer Science
2001-02	Military Service
1993-01	Federal Grammar School, (BG Dornbirn)
30. April 1983	Born in Dornbirn/Austria
Master Thesis	Recognizing Degraded Handwritten Characters (Advisor: Robert Sablatnig)
Research Interests	Document Analysis, Machine Learning, Local Features