

*Deep learning methodologies for
textual and graphical content-based
analysis of handwritten text images*

PHD THESIS

José Ramón Prieto Fontcuberta

Supervised by Prof. Emeritus Enrique Vidal

and PhD. Lorenzo Quirós Díaz

May 2024



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Work partially supported by the Universitat Politècnica de València under grant
FPI-I/900

© **José Ramón Prieto Fontcuberta, 2024**

Agradecimientos (Acknowledgements)

En este momento tan significativo de mi vida académica, me llena de alegría poder expresar mi gratitud a todas aquellas personas que han sido pilares fundamentales en este viaje.

En primer lugar, me gustaría agradecer a mi director de tesis, Enrique. Gracias por tu enseñanza y paciencia, por ayudarme a corregir mis errores y por guiarme con tanta dedicación a lo largo de este proceso.

A toda mi familia, gracias por ser un soporte incansable, por vuestro apoyo incondicional y por siempre estar allí, sin importar la distancia.

No puedo dejar de agradecer a mis compañeros del PRHLT, especialmente a Lorenzo, quien empezó como compañero y acabó siendo codirector de esta tesis. Las horas de reuniones y nuestras divagaciones sobre el futuro han sido fundamentales para mí. A Miguel y José Andrés, gracias por las incontables horas en el laboratorio y toda la ayuda proporcionada. A Vicent por iniciarme en el grupo y estar allí en mis inicios cuando lo necesitaba. Alejandro, Dani, David, Dan y todos los demás en el laboratorio, gracias por hacer este camino mucho más ameno y llevadero.

A Adrián, que desde la carrera hemos seguido el mismo camino y aún en la distancia has estado allí para hablar.

A Macarena, gracias por tu apoyo constante durante todo el doctorado, siendo un pilar crucial en mi desarrollo personal y profesional.

Claudia y Javier Fresneda, aunque la distancia nos separara, siempre he sentido vuestro apoyo y ánimo desde Menorca. A Kico y Joan, gracias por introducirme a este mundo y motivarme desde mucho antes de mi etapa universitaria.

Tatiana, gracias por tu soporte, especialmente en los momentos iniciales del doctorado, y por animarme a seguir adelante a pesar de las incertidumbres.

I would also like to thanks to Thierry and Thomas. Your warm hospitality made my time in France truly memorable and made me feel at home. Our discussions about the future implications of the research in this thesis and other fields were incredibly insightful and inspiring.

I would also like to extend my gratitude to my distant cousin Abathur for his true support throughout this journey.

Finalmente, a Antonio, Javier Monreal y Carlos, gracias por estar siempre allí para celebrar los éxitos y superar los obstáculos juntos, siempre dispuestos a terminar el día en algún bar cuando hiciera falta, compartiendo risas y desahogos.

Abstract

We are witnessing rapid advancements in Artificial Intelligence, transitioning from statistical models like Hidden Markov Models and Support Vector Machines to neural models like Convolutional Neural Networks and Transformers. These innovations have driven fields like computer vision and natural language processing to new heights. However, applying these cutting-edge techniques to the extraction and preservation of information from historical handwritten documents poses unique challenges owing to their age and degradation. While progress has been made, there remain unresolved issues that are of interest to both researchers and practitioners, including historians and paleographers.

This thesis addresses unresolved issues in the field of Artificial Intelligence as applied to historical handwritten documents. The challenges include not only the degradation of the documents but also the scarcity of available data for training specialized models. This limitation is particularly relevant when the trend is to use large datasets and massive models to achieve significant breakthroughs.

First, we provide an overview of various techniques and concepts used throughout the thesis. Different ways of representing data are explored, including images, text, and graphs. Probabilistic Indices (PrIx) are introduced for textual representation and its encoding using $Tf \cdot Idf$ is explained. We also discuss selecting the best input features for neural networks using Information Gain (IG). In the realm of neural networks, specific models such as Multilayer Perceptron (MLP), Convolutional Neural Networks (CNNs), and graph-based networks (GNNs) are covered, along with a brief introduction to transformers.

The first problem addressed in this thesis is the segmentation of historical handwritten books into semantic units, a complex and recurring challenge in archives worldwide. Unlike modern books, where chapter

segmentation is relatively straightforward, historical books present unique challenges due to their irregularities and potential poor preservation. To the best of our knowledge, this thesis formally defines this problem. We propose a pipeline to consistently extract these semantic units in two variations: one with corpus-specific constraints and another without them. Various types of neural networks are employed, including Convolutional Neural Networks (CNNs) for classifying different parts of the image and Region Proposal Networks (RPNs) and transformers for detecting and classifying regions. Additionally, a new metric is introduced to measure the information loss in the detection, alignment, and transcription of these semantic units. Finally, different decoding methods are compared, and the results are evaluated across up to five different datasets.

In another chapter, we tackle the challenge of classifying non-transcribed historical handwritten documents, specifically notarial deeds, from the Provincial Historical Archive of Cádiz. A framework is developed that employs Probabilistic Indices (PrIx) for classifying these documents, and this is compared to 1-best transcriptions obtained through Handwritten Text Recognition (HTR) techniques. In addition to conventional classification within a closed set of classes (Close Set Classification, CSC), this thesis introduces the Open Set Classification (OSC) framework. This approach not only classifies documents into predefined classes but also identifies those that do not belong to any of the established classes, allowing an expert to label them. Various techniques are compared, and two are proposed. One approach without using a threshold on the posterior probabilities generated by the neural network model. At the same time, the other employs a threshold on these probabilities, with the option for manual adjustment according to the expert's needs.

In a third chapter, this thesis focuses on Information Extraction (IE) from handwritten tabular documents. A pipeline is developed that starts with detecting text in images containing tables, line by line, followed by its transcription using HTR techniques. In parallel, various models are trained to identify the structure of the tables, including rows, columns, and header sections. The pipeline also addresses common issues in handwritten tables, such as multi-span columns and substituting ditto marks. Additionally, a language model specifically trained to detect table headers automatically is employed. Two datasets are used to demonstrate the effectiveness of the

pipeline in the IE task, and areas for improvement within the pipeline itself are identified for future research.

This thesis tackles three complex problems in the field of artificial intelligence applied to historical handwritten documents, which have been largely unexplored under the challenging conditions presented by the datasets used. The proposed solutions are significant from both a technical and practical perspective. In some cases, this is the first attempt to address these issues using historical data. Moreover, we underscore the relevance of its findings for collaborative applications with expert historians and paleographers, offering solutions to similar challenges in archives worldwide.

Resumen

Estamos experimentando rápidos avances en Inteligencia Artificial, pasando de modelos estadísticos como Hidden Markov Models y Support Vector Machines a modelos neuronales como Convolutional Neural Networks y Transformers. Estas innovaciones han impulsado campos como la visión por computadora y el procesamiento del lenguaje natural. Sin embargo, aplicar estas técnicas avanzadas a la extracción y conservación de información de documentos históricos manuscritos presenta desafíos únicos, debido a su antigüedad y degradación. Aunque se han logrado progresos, todavía hay problemas no resueltos que son de interés tanto para investigadores como para historiadores y paleógrafos.

En esta tesis se abordan problemas no resueltos en el campo de la Inteligencia Artificial aplicada a documentos históricos manuscritos. Los desafíos incluyen no solo la degradación de los documentos, sino también la escasez de datos disponibles para entrenar modelos especializados. Esta limitación es especialmente relevante en un contexto en el que la tendencia es utilizar grandes conjuntos de datos y modelos masivos para lograr avances significativos.

Primero haremos un recorrido por diversas técnicas y conceptos que se utilizarán durante la tesis. Se explorarán diferentes formas de representar datos, incluidas imágenes, texto y grafos. Se introducirá el concepto de Índices Probabilísticos (PrIx) para la representación textual y se explicará su codificación usando $Tf \cdot Idf$. También se discutirá la selección de las mejores características de entrada para redes neuronales mediante Information Gain (IG). En el ámbito de las redes neuronales, se abordarán modelos específicos como Multilayer Perceptron (MLP), Redes Neuronales Convolucionales (CNNs) y redes basadas en grafos (GNNs), además de una breve introducción a los transformers.

El primer problema que aborda la tesis es la segmentación de libros

históricos manuscritos en unidades semánticas, un desafío complejo y recurrente en archivos de todo el mundo. A diferencia de los libros modernos, donde la segmentación en capítulos es más sencilla, los libros históricos presentan desafíos únicos debido a su irregularidad y posible mala conservación. La tesis define formalmente este problema por primera vez y propone un pipeline para extraer consistentemente las unidades semánticas en dos variantes: una con restricciones del corpus y otra sin ellas. Se emplearán diferentes tipos de redes neuronales, incluidas CNNs para la clasificación de partes de la imagen y RPNs y transformers para detectar y clasificar regiones. Además, se introduce una nueva métrica para medir la pérdida de información en la detección, alineación y transcripción de estas unidades semánticas. Finalmente, se comparan diferentes métodos de “decoding” y se evalúan los resultados en hasta cinco conjuntos de datos diferentes.

En otro capítulo, la tesis aborda el desafío de clasificar documentos históricos manuscritos no transcritos, específicamente actos notariales en el Archivo Provincial Histórico de Cádiz. Se desarrollará un framework que utiliza Índices Probabilísticos (PrIx) para clasificar estos documentos y se comparará con transcripciones 1-best obtenidas mediante técnicas de Reconocimiento de Texto Manuscrito (HTR). Además de la clasificación convencional en un conjunto cerrado de clases (Close Set Classification, CSC), la tesis introduce el framework de Open Set Classification (OSC). Este enfoque no solo clasifica documentos en clases predefinidas, sino que también identifica aquellos que no pertenecen a ninguna de las clases establecidas, permitiendo que un experto los etiquete. Se compararán varias técnicas para este fin y se propondrán dos. Una sin umbral en las probabilidades a posteriori generadas por el modelo de red neuronal, y otra que utiliza un umbral en las mismas, con la opción de ajustarlo manualmente según las necesidades del experto.

En un tercer capítulo, la tesis se centra en la Extracción de Información (IE) de documentos tabulares manuscritos. Se desarrolla un pipeline que comienza con la detección de texto en imágenes con tablas, línea por línea, seguido de su transcripción mediante técnicas de HTR. De forma paralela, se entrenarán diferentes modelos para identificar la estructura de las tablas, incluidas filas, columnas y secciones de cabecera. El pipeline también aborda problemas comunes en tablas manuscritas, como el multi-span de

columnas y la sustitución de texto entre comillas. Además, se emplea un modelo de lenguaje entrenado específicamente para detectar automáticamente las cabeceras de las tablas. Se utilizarán dos conjuntos de datos para demostrar la eficacia del pipeline en la tarea de IE, y se identificarán las áreas de mejora en el propio pipeline para futuras investigaciones.

La tesis aborda tres problemas complejos en el campo de la inteligencia artificial aplicada a documentos históricos manuscritos, que hasta ahora han sido poco explorados en las condiciones desafiantes presentadas por los datasets utilizados. Las soluciones propuestas son significativas tanto desde una perspectiva técnica como práctica. En algunos casos, se trata de la primera vez que se intenta resolver estos problemas con datos históricos. Además, la tesis destaca la relevancia de sus hallazgos para aplicaciones en colaboración con expertos historiadores y paleógrafos, ofreciendo soluciones a problemas similares en archivos de todo el mundo.

Resum

Estem experimentant avanços ràpids en Intel·ligència Artificial, passant de models estadístics com ara Hidden Markov Models i Support Vector Machines a models neuronals com ara Convolutional Neural Networks i Transformers. Aquestes innovacions han impulsat camps com la visió per computadora i el processament del llenguatge natural. No obstant això, aplicar aquestes tècniques avançades a l'extracció i conservació d'informació de documents històrics manuscrits presenta desafiaments únics, degut a la seva antiguitat i degradació. Tot i que s'han aconseguit progressos, encara hi ha problemes no resolts que són d'interès tant per a investigadors com per a historiadors i paleògrafs.

En aquesta tesi s'aborden problemes no resolts en el camp de la Intel·ligència Artificial aplicada a documents històrics manuscrits. Els desafiaments inclouen no només la degradació dels documents, sinó també l'escassetat de dades disponibles per entrenar models especialitzats. Aquesta limitació és especialment rellevant en un context en què la tendència és utilitzar grans conjunts de dades i models massius per aconseguir avanços significatius.

Primer farem un recorregut per diverses tècniques i conceptes que s'utilitzaran durant la tesi. S'exploraran diferents formes de representar dades, incloses imatges, text i grafos. S'introduirà el concepte d'Índexs Probabilístics (PrIx) per a la representació textual i s'explicarà la seva codificació usant $Tf \cdot Idf$. També es discutirà la selecció de les millors característiques d'entrada per a xarxes neuronals mitjançant Information Gain (IG). En l'àmbit de les xarxes neuronals, s'abordan models específics com Multilayer Perceptron (MLP), Xarxes Neuronals Convolucionals (CNNs) i xarxes basades en grafos (GNNs), a més d'una breu introducció als transformers.

El primer problema que aborda la tesi és la segmentació de llibres

històrics manuscrits en unitats semàntiques, un desafiament complex i recurrent en arxius de tot el món. A diferència dels llibres moderns, on la segmentació en capítols és més senzilla, els llibres històrics presenten desafiaments únics degut a la seva irregularitat i possible mala conservació. La tesi defineix formalment aquest problema per primera vegada i proposa un pipeline per extreure consistentment les unitats semàntiques en dues variants: una amb restriccions del corpus i una altra sense elles. S'empraran diferents tipus de xarxes neuronals, incloses CNNs per a la classificació de parts de la imatge i RPNs i transformers per detectar i classificar regions. A més, s'introdueix una nova mètrica per mesurar la pèrdua d'informació en la detecció, alineació i transcripció d'aquestes unitats semàntiques. Finalment, es compararan diferents mètodes de “decoding” i s'avaluaran els resultats en fins a cinc conjunts de dades diferents.

En un altre capítol, la tesi aborda el desafiament de classificar documents històrics manuscrits no transcrits, específicament actes notarials a l'Arxiu Provincial Històric de Càdiz. Es desenvoluparà un marc que utilitza Índexs Probabilístics (PrIx) per classificar aquests documents i es compararà amb transcripcions 1-best obtingudes mitjançant tècniques de Reconèixer Text Manuscrit (HTR). A més de la classificació convencional en un conjunt tancat de classes (Close Set Classification, CSC), la tesi introdueix el marc d'Open Set Classification (OSC). Aquest enfocament no només classifica documents en classes predefinides, sinó que també identifica aquells que no pertanyen a cap de les classes establertes, permetent que un expert els etiqueti. Es compararan diverses tècniques per a aquest fi i es proposaran dues. Una sense llinar en les probabilitats a posteriori generades pel model de xarxa neuronal, i una altra que utilitza un llinar en les mateixes, amb l'opció d'ajustar-lo manualment segons les necessitats de l'expert.

En un tercer capítol, la tesi es centra en l'Extracció d'Informació (IE) de documents tabulars manuscrits. Es desenvolupa un pipeline que comença amb la detecció de text en imatges amb taules, línia per línia, seguit de la seva transcripció mitjançant tècniques de HTR. De forma paral·lela, s'entrenaran diferents models per identificar l'estructura de les taules, incloses files, columnes i seccions de capçalera. El pipeline també aborda problemes comuns en taules manuscrites, com ara el multi-span de columnes i la substitució de text entre cometes. A més, s'empra un model de llenguatge entrenat específicament per detectar automàticament les capçaleres

de les taules. S'utilitzaran dos conjunts de dades per demostrar l'eficàcia del pipeline en la tasca de IE, i s'identificaran les àrees de millora en el propi pipeline per a futures investigacions.

La tesi aborda tres problemes complexos en el camp de la intel·ligència artificial aplicada a documents històrics manuscrits, que fins ara han estat poc explorats en les condicions desafiantes presentades pels datasets utilitzats. Les solucions proposades són significatives tant des d'una perspectiva tècnica com pràctica. En alguns casos, es tracta de la primera vegada que s'intenta resoldre aquests problemes amb dades històriques. A més, la tesi destaca la rellevància dels seus resultats per a aplicacions en col·laboració amb experts historiadors i paleògrafs, oferint solucions a problemes similars en arxius de tot el món. En resum, la tesi contribueix de manera significativa al camp de la Intel·ligència Artificial aplicada a documents històrics manuscrits, i obre noves vies per a futures investigacions i aplicacions pràctiques en aquest àmbit.

Preface

The task of preserving and interpreting our historical heritage is of incalculable value. Historical documents serve as time capsules, housing a wealth of knowledge, narratives, and cultural details. However, the delicate and intricate nature of these manuscripts poses considerable challenges, especially when it comes to information extraction, classification, and preservation. In this context, the disciplines of Computer Vision (CV) and Natural Language Processing (NLP) offer a promising toolkit for overcoming these challenges.

The main goal of this thesis is to delve into several uncharted challenges associated with the analysis and processing of historical handwritten documents. While there has been significant research in this domain, numerous unanswered questions and unresolved problems still exist, justifying the need for a more focused and specialized research.

Specifically, we focus on three fundamental areas:

- **Book Segmentation:** We explored various algorithms and methodologies to effectively segment historical books into semantic units, such as chapters or legal deeds. This is essential for easing further analysis and interpretation.
- **Document Classification:** We developed advanced classification techniques tailored for historical documents that lack transcription. The objective is to accurately identify the document types, a critical step for proper archiving and study.
- **Information Extraction in Tabular Data:** We explored the applicability of modern data extraction techniques to interpret and store information presented in tabular formats within these documents.

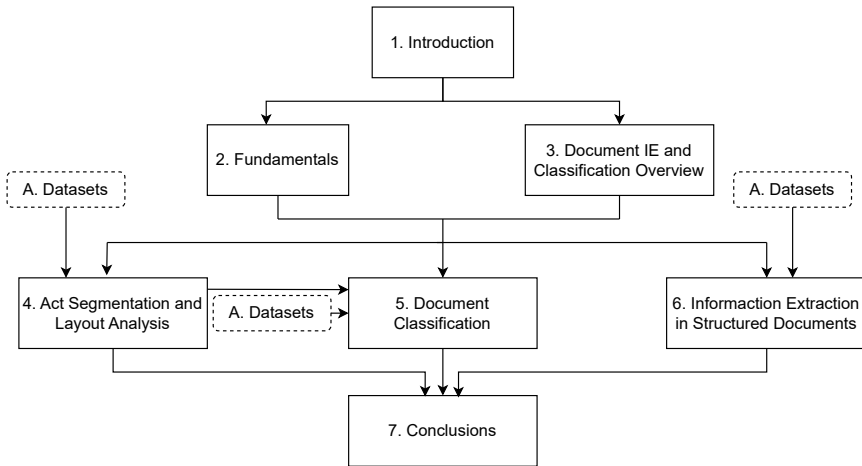


Figure 1: Dependency diagram between the chapters of this thesis.

The intersection of these three core areas allows for a more comprehensive and robust approach to understanding and preserving historical handwritten documents. With this, we aim to contribute to the interdisciplinary field that spans technology and the humanities, thereby enabling the conservation of our historical heritage.

In order to explain and evaluate the proposed approaches, this thesis is divided into seven chapters and an appendix. We recommend that readers follow this sequential order. However, some chapters can be read out of order or skipped, depending on the reader’s interest. In this case, we provide a dependency diagram between chapters in Figure 1.

The content of each chapter is as follows:

Chapter 1: The opening chapter serves as an introduction to the key issues that the thesis explore, with a particular focus on the significance of historical documents.

Chapter 2: This chapter aims to provide a theoretical background on the methods and algorithms that are used throughout this thesis. If the reader is already familiar with the subject matter, this chapter can be skipped.

Chapter 3: In this chapter, we revisit the importance of the analysis and classification of historical documents. We focus on showcasing the difficulties and challenges we face when working with these types of documents.

Chapter 4: In this chapter, we tackle the problem of book segmentation. We provide a formal definition of the problem and approach it from various angles. We also introduce a new metric to measure the results of segmentation and joint transcription. The results on various corpora with different constraints are shown.

Chapter 5: In this chapter, we provide an effective solution for classifying handwritten historical documents using the Open Set Classification (OSC) framework. Additionally, we also discuss how to automatically calculate the threshold for rejecting samples from unseen classes. Finally, we use probabilistic indices and compare them with plain text.

Chapter 6: In this chapter, we address the problem of information extraction in historical tabular data. A pipeline is created where starting from the image, we use different techniques to extract the text, transcribe it, and obtain the table structure. Finally, we are able to extract the information.

Chapter 7: This chapter summarizes all the contributions of this work, including scientific publications and projects in which participation has occurred. A perspective on possible future work in each of the open research avenues is provided.

Appendix: In the appendix, we find the description and statistics of all the datasets used during the thesis.

Abbreviations

AHPC Spanish Provincial Historical Archive of Cádiz

ANN Artificial Neural Network

BoW Bag of Words

BSER Bundle Segmentation Error Rate

CAER Content Alignment and Error Rate

CBIDC Content Based Image Document Classification

CC Consistency Constraints

C Complete Act

CE Cross-Entropy

CER Character Error Rate

CNN Convolutional Neural Network

CSC Closed Set Classification

CV Computer Vision

DC Document Classification

DLA Document Layout Analysis

F Final Act

FN False Negative

FP False Positive

GNN Graph Neural Network

GT Ground Truth

HMM Hidden Markov Model

HTR Optical Character Recognition

HWR Handwriting Recognition

Idf Inverse Document Frequency

IE Information Extraction

IG Information Gain

I Initial Act

IoU Intersection over Union

mAP mean Average Precision

ML Machine Learning

MLP Multilayer Perceptron

M Medium Act

MP Message-Passing

MPNN Message Passing Neural Networks

NER Named Entity Recognition

NLP Natural Language Processing

OCR Handwritten Text Recognition

OSC Open Set Classification

PrIx Probabilistic Indexing

PR Pattern Recognition

RCSA Royal Cedulae Simancas Archive
ReLU Rectified Linear activation function
RJ Reject
RNN Recurrent Neural Network
RPN Region Proposal Network
RP Relevance Probability
SGD Stochastic Gradient Descent
SVM Support Vector Machines
Tf·Idf Term Frequency-Inverse Document Frequency
Tf Term Frequency
ToC Table of Content
TP True Positive
WER Word Error Rate

Contents

Abbreviations	xix
Contents	xxiii
1 Introduction	1
2 Fundamentals	5
2.1 Data Representation	5
2.2 Artificial Neural Networks	12
2.3 Open and Closed Set Classification	25
3 Document Information Extraction And Classification Overview	29
3.1 Document Layout: Transcending Single Pages . . .	34
3.2 Document Classification of Historical Manuscripts .	38
3.3 Information Extraction in Historical Manuscripts . .	43
4 Act Segmentation and Layout Analysis	49
4.1 Problem Definition	50
4.2 A Whole-Book Evaluation Measure	57
4.3 Restricted Multi-page Act Segmentation	60
4.4 Multi-page Act Fine-Grained Segmentation	71
4.5 Simancas Archive Segmentation	92
4.6 Discussion	98
5 Document Classification	103
5.1 Problem Definition	105
5.2 Feature Selection and Extraction for CBIDC	108
5.3 Open Set Classification in AHPC	109
5.4 Discussion	119

Contents

6	Information Extraction in Structured Documents	121
6.1	Problem Definition	122
6.2	Evaluation Measures	128
6.3	Information Extraction in HisClima Tables	130
6.4	Discussion	141
7	Conclusions	143
7.1	Scientific Outcomes	145
7.2	Projects	148
7.3	Open Source Software	148
7.4	Future Work	149
	Appendices	153
A	Datasets	155
A.1	Alcar - HOME	155
A.2	Archivo Histórico Provincial de Cádiz (AHPC)	158
A.3	Hisclima	162
A.4	RCSA Dataset	165
	List of Figures	169
	List of Tables	171
	Bibliography	173

1 *Introduction*

Since the earliest times of human civilization, the imperative to record and disseminate knowledge has been a driving force behind the development of various documentation techniques and mediums. The earliest efforts to immortalize information were cave paintings and stone carvings. While these methods were primitive, they effectively served as a means of communication within communities. These ancient inscriptions, discovered in locations ranging from the Lascaux caves in France to Altamira in Spain, can be seen as the precursors to modern documentation, even though they do not constitute a writing system in the modern sense of the term.

With the emergence of the first civilizations, the art of writing became a fundamental tool for the administration and organization of increasingly complex societies. For instance, the invention of cuneiform writing in Mesopotamia between 3500 and 3000 B.C. represented a pivotal advancement in the history of documentation systems [Que05]. Inscribed on clay tablets, these early scripts covered various subjects – from laws and commercial transactions to inventory lists, medical prescriptions, and mythological stories. This rich tapestry of written records offers an invaluable window into the complexities and preoccupations of ancient civilizations.

The medium on which writing was done also underwent several transformations throughout history. In ancient Egypt, the use of papyrus allowed for greater portability and storage of documents, which resulted in the creation of some of the world’s oldest libraries. Iconic documents like the “Book of the Dead” have reached us thanks to the durability of papyrus as a means of documentation [Par19].

Subsequently, the invention of paper in China in the 2nd century AD opened up new possibilities for producing and distributing texts. As the paper-making technique spread worldwide, it became the most popular medium for writing, both for official records and for literary and academic

1. Introduction

works.

During the Middle Ages, monasteries acted as epicenters of scholarship and manuscript production in Europe. Monks took on the laborious task of transcribing texts, often adding illustrations and ornamentations that transformed each manuscript into a unique work of art [Dom14]. Although many of these manuscripts were of a religious nature, works by ancient philosophers, medical treatises, and legal texts were also copied.

The arrival of the movable-type printing press in the 15th century represented a true revolution in how documents were produced and distributed. This invention, attributed to Johannes Gutenberg, allowed for the mass production of texts, thus democratizing access to knowledge and forever changing the dynamics of learning and the dissemination of ideas [Gil96].

Despite the technological advancements in document production, handwritten historical documents present their own unique set of challenges and significance. Unlike printed materials, which are generally uniform and easier to read, manuscripts can vary widely regarding handwriting style, ink quality, and preservation quality. Many of these invaluable documents have yet to be transcribed or digitized, rendering them largely inaccessible to the general public. The monumental task of classifying, extracting information from, and segmenting these manuscripts calls for a multidisciplinary approach that combines expertise in history, paleography, and increasingly cutting-edge technologies like artificial intelligence [Noc+22; MLK20].

In the 1950s, the first feedforward neural networks emerged, although they were not yet categorized under the umbrella of “Deep Learning”. This period also marked the birth of a new research field known as Optical Character Recognition (OCR), which focuses on identifying characters within images. The field has seen significant advancements over the years, including the development of backpropagation training methods, the introduction of convolutional neural networks, generative adversarial networks, recurrent neural networks, and the use of the power of GPUs for training these complex models [Sch22].

With advancements in OCR, a new specialized field emerged known as Handwritten Text Recognition (HTR). Unlike OCR, which focuses on printed text, HTR is dedicated to recognizing handwritten text. This area of study becomes particularly challenging when dealing with historical handwritten documents, given their unique difficulties and peculiarities.

Before the widespread adoption of deep learning technologies, various methods were employed for text recognition and document classification. One of the most notable was the Hidden Markov Model (HMM). These models relied on statistical and probabilistic methods and were extensively used in voice and handwriting recognition tasks due to their ability to manage temporal data sequences [Cam20; FSV02].

Along with HMMs, Support Vector Machines (SVM) also played a crucial role in classification. Although effective in their time, these models had limitations in terms of processing capacity and adaptability to new data [AK15].

The advent of deep learning has resulted in a renaissance of artificial intelligence and pattern recognition. Convolutional Neural Networks (CNNs) have been game-changers in the realms of image and text recognition. Inspired by the human brain's architecture, CNNs can process images and text with remarkable accuracy, eliminating the need for explicit feature extraction commonly required in traditional methods [Alz+21].

Recurrent Neural Networks (RNNs) have become the go-to solution for tasks involving sequences like handwritten text. These networks can “remember” information from previous steps, making them particularly well-suited for tasks such as transcribing handwritten text [GS08].

More sophisticated models like Convolutional Recurrent Neural Networks (CRNNs) were developed as technology evolved. These models combine the strengths of both CNNs and RNNs, offering more efficient processing of images and text [Pui18].

In the modern era, end-to-end models have become popular, allowing direct processing from input to output without intermediate stages [CCP23], since until now, work was done at the line level, having first to perform line extraction. In addition, Transformers, which use attention mechanisms to weigh the relative importance of different parts of the data, have set new standards in tasks such as machine translation and text processing, among others [Isl+23].

Even with the significant advancement of deep learning, there is still a long way to go in the field of historical handwritten documents. Much progress has been made regarding text recognition and document layout analysis [Qui22]. However, there is still a long way to go due to the incredible complexity and the many peculiarities that arise from such documents.

1. Introduction

Many tasks still need to be addressed, among other reasons, due to the lack of data or resources.

This thesis embarks on an interdisciplinary journey to address a series of challenges inherent to historical handwritten documents. Specifically, the following objectives have been established:

Historical Book Segmentation: Address the complex task of segmenting notarial acts within historical books using artificial intelligence techniques. This objective considers the various contexts and challenges currently present in archival settings. Successful segmentation not only makes these records more accessible but also better organized, thereby paving the way for future information extraction efforts.

Classification of Untranscribed Documents: Develop models that can accurately classify untranscribed historical handwritten documents. There are thousands of documents, which can be up to hundreds of pages each, where transcribing them with any degree of accuracy is challenging due to the conditions in which they are found and where transcription error rates are exceedingly high, even with the latest HTR techniques.

Information Extraction in Documents: Design robust systems that go beyond simple transcription in structured documents, such as tables or forms in historical documents. These systems should be capable of identifying and extracting structured information, such as dates, names, places, and events, allowing for a deeper understanding of the content and facilitating its analysis and study.

As we embark on this thesis, we recognize the magnitude and complexity of the challenges we face, some of which have not been addressed before. However, preserving, understanding, and sharing our documentary heritage cannot be underestimated. Each manuscript, each page, and each word are silent witnesses to our history, and we must give them a voice in the digital age.

We aspire not only to advance in the academic and technological field but also to build bridges between the past and the present. In the end, we hope to have contributed in some way to deciphering, classifying, and preserving the hidden treasures in our historical handwritten documents, ensuring that their legacy endures for future generations.

Fundamentals

2

In this chapter, several fundamental techniques, methods, and algorithms used throughout the thesis are presented along with their key features. However, it is important to notice that this overview is not intended to be comprehensive, and readers who wish to delve deeper into the subject matter are advised to consult the sources listed in the bibliography.

2.1 Data Representation

Data representation is an essential aspect of machine learning, as it determines how the algorithms will process and learn from the information.

The data modality used depends on the task at hand and may include images, text, audio signals, among other types of data. Sometimes, a problem involves multiple types of data, making it a multimodal problem. Once the data modality is selected, it must be represented. For example, an image can be represented as a vector of numerical values, with each value representing a pixel in a certain position. Text can be represented using a bag-of-words model or word embeddings, which are vector representations of the data with different contexts and/or sizes. In this thesis, we will also employ graph representations, which can be used to represent any type of data modality and allow for the addition of special features such as node and edge properties. Furthermore, graph techniques and post-processing methods can be applied to the data.

The goal is to choose a representation that captures the relevant information and allows the machine learning algorithm to learn patterns and make accurate predictions.

2. Fundamentals

2.1.1 Image

Images are typically encoded as arrays of numerical values, which can be processed by algorithms to perform tasks such as object recognition, image classification, and image segmentation.

First, the image is captured using a camera or other imaging device. Then, the image may be preprocessed to enhance its quality, reduce noise or be able to fit it into a model. For instance, by applying filters, adjusting brightness and contrast, or resizing the image. However, these preprocessing steps are being less necessary thanks to the advances in the Computer Vision (CV) field. Finally, the image is then encoded as an array of numerical values, where each pixel is assigned a value that represents its intensity or color. For grayscale images, each pixel is assigned a single value representing its brightness, usually ranging from 0 (black) to 255 (white), as can be seen in Figure 2.1. For color images, each pixel is represented by three values, one for each color channel (red, green, and blue), typically ranging from 0 to 255. The numerical values in the array may be normalized to a common scale, such as by dividing each pixel value by 255 to obtain values between 0 and 1.

It is important to notice that images can be represented in various ways. However, in the context of this thesis, we are particularly interested in any approach which utilizes normalized values. This is because models such as neural networks, which are extensively used in this thesis, tend to perform better with this type of representation.

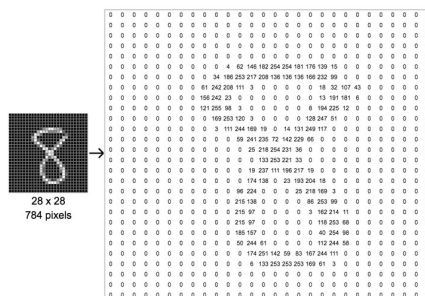


Figure 2.1: Example of a handwritten number 8 in grayscale represented as a matrix.

Once the image is encoded, it can be used as input to machine learning algorithms such as CNNs, which can learn to recognize patterns in the

image and make predictions.

2.1.2 Text

Text data is encoded as a numerical vector which can be processed by algorithms to perform tasks such as sentiment analysis, text classification, and language modeling.

The process of encoding text involves several steps. First, tokenization is done by breaking down the text into individual tokens, such as words, subwords or characters, which are then represented as discrete units that can be processed by machine learning algorithms. Then, a vocabulary is created that maps each token to a unique integer index. The size of the vocabulary will depend on the number of distinct tokens in the text data. Each token in the text is then replaced by its corresponding index in the vocabulary, resulting in a sequence of integers. This sequence can be further transformed into a numerical vector, such as a bag-of-words (BoW) model [IKT05; MRS08; AZ12], where each dimension of the vector corresponds to a distinct token in the vocabulary, and the value of each dimension represents the frequency of that token in the text, ignoring the order of words.

Another transformation can be applied, such as word or subword embeddings, which represent each token as a dense, low-dimensional vector that captures semantic meaning and relationships between words. Usually, and more common in BoW models, resultant numerical values in the vector are normalized to a common scale, such as by dividing each value by the total number of tokens in the text.

Once the text is encoded, it can be used as input to machine learning algorithms such as Multilayer Perceptron (MLP) or recurrent neural networks (RNNs), which can learn to recognize patterns in the text and make predictions.

2.1.2.1 Term Frequency - Inverse Document Frequency

Term Frequency-Inverse Document Frequency (Tf-Idf) is a numerical statistic that aims to reflect the importance of a word within a document in the context of a larger corpus. This method has been widely used in

2. Fundamentals

information retrieval and text mining as a weighting factor during indexing, ranking, and document similarity calculations.

For a specific document D , a model takes an input feature vector $\vec{D} \in \mathbb{R}^n$, where n represents the number of features from the overall vocabulary $v \in V$. The value D_v of each word v is generally associated with the number of occurrences of v in D . Then, let $f(v, D)$ be the number of occurrences of v in D . One possible definition for D_v could be $D_v = f(v, D)$. However, the absolute number of word occurrences can vary significantly with document size. Let $f(D) = \sum_{v \in V} f(v, D)$ denote the total (or “running”) number of words in D . Consequently, for each $v \in V$, the normalized frequency $f(v, D) / f(D)$ is generally preferred. This ratio, denoted as $\text{Tf}(v, D)$ and often referred to as *term frequency*, serves as a maximum-likelihood estimate of the conditional probability of word v , given a document D , $P(v | D)$.

Although Tf effectively addresses variations in document size, it is suggested that improved document classification accuracy can be obtained by assigning additional weights to each feature based on their predictive “importance” for determining the document’s class. While techniques like Information Gain could be employed for this purpose, the *inverse document frequency* (Idf) [SB88; Joa96; Aiz03] is considered more suitable. With $f(v) \leq M \stackrel{\text{def}}{=} |\mathcal{D}|$ being the number of documents in the set of documents \mathcal{D} which contain v and M defined as the total number of documents, Idf is defined as $\log(M / f(v))$.

Finally, to construct a feature vector \vec{D} for a document D , the value of each feature, D_v , is calculated as the product of $\text{Tf}(v, D)$ and $\text{Idf}(v)$:

$$\begin{aligned}
D_v &= \text{Tf} \cdot \text{Idf}(v, D) \\
&= \text{Tf}(v, D) \cdot \text{Idf}(v) \\
&= P(v|D) \log \frac{M}{f(v)} \\
&= \frac{f(v, D)}{f(D)} \log \frac{M}{f(v)}
\end{aligned}
\tag{2.1}$$

As a result, words that frequently occur in a specific document but infrequently throughout the entire corpus will have high Tf-Idf values, signifying their significance within that specific document.

In summary, Tf-Idf is a widely used technique in text analysis and information retrieval to quantify the relevance of terms within documents and across a corpus. This method helps to filter out common words with little informational value and highlight words that carry significant meaning in the context of the given document.

2.1.2.2 Information Gain

Not all words in a document D are equally helpful for describing or predicting its class. A common first step in document classification is determining a “useful” vocabulary, denoted as V_n , with a reasonable size $n < N$, where N is the total size of the vocabulary. One of the most effective methods to determine such a vocabulary is by calculating the Information Gain (IG) of each word that appears in the document set \mathcal{D} and selecting the top n words with the highest IG to be included in V_n . IG is a metric used to select features - in this case, words - for document classification. It is a way to rank the words in the vocabulary based on how well they differentiate or classify documents into different classes.

This method begins by defining the value of a boolean random variable, t_v , for each word v . If a randomly selected document D contains the word v , t_v is True; otherwise, it is False. Here $P(T_v = 1)$ is the probability that a document contains the word v , and $P(T_v = 0)$ is the probability that a document does not contain v . Nevertheless, for the sake of simplicity

2. Fundamentals

henceforth, we shall denote $P(T_v = 1)$ as $P(t_v)$, and $P(T_v = 0)$ shall be represented as $P(\bar{t}_v)$.

The IG of a word v is then computed using a formula that measures the difference between the entropy of the entire dataset (how uncertain we are about the classification of a randomly selected document) and the weighted sum of the entropies of the subsets of documents that contain v and that do not contain v . Each entropy is calculated using the probabilities of each class c .

The formula for Information Gain $IG(v)$ is:

$$\begin{aligned} IG(v) = & - \sum_{c \in \mathcal{C}} P(c) \log P(c) \\ & + P(t_v) \sum_{c \in \mathcal{C}} P(c | t_v) \log p(c | t_v) \\ & + P(\bar{t}_v) \sum_{c \in \mathcal{C}} P(c | \bar{t}_v) \log P(c | \bar{t}_v) \end{aligned} \quad (2.2)$$

where:

- $P(c)$ is the prior probability of class c (the probability of any document belonging to class c , without any other information).
- $P(c | t_v)$ is the conditional probability of a document belonging to class c , given that it contains the word v .
- $P(c | \bar{t}_v)$ is the conditional probability of a document belonging to class c , given that it does not contain v .
- \mathcal{C} is the set of classes.

The first term in the equation is constant for all words; therefore, it can be ignored when comparing words.

Thus, the probability that there exists a document $D \in \mathcal{D}$ in which the word v is written is denoted as $P(t_v)$,

while $P(\bar{t}_v)$ is the probability that the word v is not written.

Then, we can calculate the required probabilities for IG as follows:

$$P(t_v) = \frac{m(v, \mathcal{D})}{M}, \quad P(c | t_v) = \frac{m(v, \mathcal{D}_c)}{m(v, \mathcal{D})} \quad (2.3)$$

$$P(\bar{t}_v) = 1 - P(t_v), \quad P(c | \bar{t}_v) = \frac{M_c - m(v, \mathcal{D}_c)}{M - m(v, \mathcal{D})}$$

where \mathcal{D}_c is a subset of \mathcal{D} where all documents belongs to class c , and M_c is the number of documents in the subset \mathcal{D}_c , as well as M is the number of documents on \mathcal{D} . Then, $m(v, \mathcal{D})$ is the number of documents from \mathcal{D} that contain the word v .

2.1.3 Graph-based representations

Graphs are used to represent the relationships between entities, providing a universal language for representing and analyzing complex systems. The data is typically encoded as a collection of nodes connected by edges, which can be processed by algorithms to perform tasks such as node classification, graph classification, and link prediction.

A graph is represented $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where \mathcal{V} is a set of nodes, and \mathcal{E} is a set of edges. We will denote a node as $v_i \in \mathcal{V}$ and $e_{ij} \in \mathcal{E}$ an edge pointing from node v_i to node v_j . From the set of edges, we obtain an adjacency $|\mathcal{V}| \times |\mathcal{V}|$ binary matrix \mathcal{A} , with a_{ij} as:

$$a_{ij} = \begin{cases} 1 & e_{ij} \in \mathcal{E} \\ 0 & e_{ij} \notin \mathcal{E} \end{cases} \quad (2.4)$$

When a graph has node attributes is referred to as an *attributed graph* in the literature, where $\mathcal{X} \in \mathbb{R}^{|\mathcal{V}|, d}$ is the node feature matrix, with $\mathcal{X}_i \in \mathbb{R}^d$ is representing the feature vector of a node v_i with d features. Meanwhile, a graph may have also edge attributes, where the $\mathcal{T} \in \mathbb{R}^{|\mathcal{E}|, d'}$ is the edge feature matrix, with $\mathcal{T}_{ij} \in \mathbb{R}^{d'}$ is representing the feature vector of an edge v_{ij} with a d' features.

The previous definition of a graph refers to a *directed graph*, where \mathcal{A} is not symmetric, and the edges do not always have to go in both directions. We will also consider a *undirected graph*, which is a particular case of graphs where the adjacency matrix \mathcal{A} is symmetric, and therefore there is a pair of edges with inverse directions if two nodes are connected.

In conclusion, each node in the graph is represented as a feature vector, encompassing attributes such as its degree, centrality, or associated label. For instance, a node might represent a character, a word, or a line of text, and its features could include its geometry and textual content.

2. Fundamentals

Furthermore, each edge represents the tuple of connected nodes v_i and v_j . These edges can also possess vector-represented features, such as weight or direction.

2.2 Artificial Neural Networks

An artificial neural network (ANN) is a computational model inspired by the structure and function of the human brain. It consists of a collection of interconnected processing units, called neurons, that are organized into layers. Each neuron receives input from other neurons or from external sources, processes that input using an activation function, and produces an output that is transmitted to other neurons.

The basic building block of an artificial neuron is the weighted sum function, which computes the sum of the products of the input values, \vec{x} , and their corresponding weights, \vec{w} , plus an independent term called bias, b , where \vec{w} and b are englobed in the parameters of the network, $\vec{\Theta}$:

$$f(\vec{x}, \vec{\Theta}) = \sum_{i=1}^{|\vec{x}|} w_i x_i + b \quad (2.5)$$

This equation forms a single unit neuron called *Perceptron* [Ros58]. Multiple units can be organized into layers, with each layer receiving input from the previous layer and producing output that is transmitted to the next layer. We will refer to the first layer as the input layer, and the last layer as the output layer. All of the layers in between will be referred to as hidden layers. If the model follows a series of layers composed of artificial neurons as shown in Eq. (2.5) it is referred to as a fully connected multilayer network, since each neuron in a layer is connected to all the neurons in the next layer. As previously mentioned, when we lack hidden layers, this model is called *Perceptron*, meanwhile when we have one or more hidden layers is usually called *Multilayer Perceptron* (MLP).

Then, the output of a layer is passed through an activation function, which introduces nonlinearity into the model and enables it to capture complex patterns in the data. Historically, non-linear activation functions were a big breakthrough in neural networks because they became capable of modeling a wide range of complex relationships between inputs and outputs.

They also enable a two-layer neural network to be a universal function approximator [HSW89]. One of the most commonly used activation functions is the sigmoid function, which maps the weighted sum to a value between 0 and 1:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (2.6)$$

The sigmoid function can be visualized as an S-shaped curve that saturates at the extremes. Another common activation function is the Rectified Linear activation function (ReLU) [Aga18], which is usually the default activation function recommended for use with most of the ANNs [GBC16]. The ReLU can be defined as the following equation, where if the input z is negative, then $\sigma(z)$ is equals to zero; otherwise, $\sigma(z)$ is z .

$$\sigma(z) = \max(0, z) \quad (2.7)$$

The weights of the connections between neurons are learned from data using an optimization algorithm, such as stochastic gradient descent (SGD), that minimizes a loss function that measures the difference between the predicted output and the true output.

When dealing with classification problems, ANNs are trained to generate outputs that reflect the posterior probability of a label (i.e., the class) based on the input data. In such cases, the softmax function [Bri90] is used

$$y_i = \sigma(z_1, \dots, z_n) = \frac{\exp(z_i)}{\sum_{j=1}^n \exp(z_j)} \quad (2.8)$$

where y_i is the i -th output neuron of the ANN.

An ANN can be used for a wide range of tasks, such as image and speech recognition, natural language processing, and predictive modeling. The choice of architecture, activation function, and optimization algorithm will depend on the specific problem being addressed and the characteristics of the data.

2.2.1 Training Process

When training a neural network, the objective is to adjust the model's parameters $\vec{\Theta}$ to minimize the difference between the predicted output and the actual output for a given set of input data. This process can be framed as

2. Fundamentals

an optimization problem, where the goal is to adjust the set of parameters $\vec{\Theta}$ that minimize a certain cost or loss function $J(\vec{\Theta})$. The training process is iterative, and at each iteration, the model's parameters $\vec{\Theta}$ are updated in the opposite direction of the gradient of the cost function $\Delta J(\vec{\Theta})$ w.r.t. the parameters.

Stochastic Gradient Descent (SGD) [BB07] is a popular optimization algorithm used for this purpose, where the parameters $\vec{\Theta}$ are updated based on the gradient's direction.

Then, *back propagation* [RHW86] is a common algorithm to combine with SGD. It is used to compute the gradient of the loss function concerning the model's parameters. It works by recursively applying the chain rule of calculus to compute the gradient of the loss function w.r.t each parameter in the model.

SGD, similar to other optimization methods, depends on the cost function to guide the optimization process. It is therefore crucial to define a cost function that aligns with the model's objectives and aids the optimizer's work as much as possible, such as being globally continuous and differentiable. However, it is often not feasible to define a cost function with all the desirable mathematical properties. In such cases, the suitability of the chosen function should be evaluated empirically.

As an example, consider a binary classification problem, where the output of an ANN with a logistic sigmoid activation function on the output layer is represented by $\sigma(\vec{x}, \vec{\Theta})$, where $\sigma : \mathbb{R}^d \rightarrow (0, 1)$. In this context, $\sigma(\vec{x}, \vec{\Theta})$ denotes the conditional probability of the target $y \in \{0, 1\}$ given the input $\vec{x} \in \mathbb{R}^d$, subject to the model parameters $\vec{\Theta}$.

The objective is to estimate the parameters $\vec{\Theta}$ that minimize the expected dissimilarity between the empirical distribution, defined by the training data, and the model distribution, measured using the Kullback-Leibler (KL) divergence. However, it has been shown [Bis07] that minimizing this KL divergence is equivalent to minimizing a cross-entropy (CE) cost function between these distributions, which can be defined as:

$$J(\vec{\Theta}) = - \sum_{n=1}^N y_n \log(\sigma(\vec{x}_n, \vec{\Theta})) + (1 - y_n) \log(1 - \sigma(\vec{x}_n, \vec{\Theta})) \quad (2.9)$$

where N is number of samples of the training set $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$ with target values represented as one-hot $Y = \{y_1, y_2, \dots, y_N\}$.

In the case of the classical multinomial classification problem, where each input is assigned to one of K mutually exclusive classes and a softmax activation function is used on the output layer of the ANN, it is typical to normalize the cost function in terms of the number of samples (N) and predictions performed (C). This leads us to a version of the optimization criterion expressed as an expectation w.r.t the empirical distribution defined by the training data [GBC16]:

$$J(\vec{\Theta}) = -\frac{1}{N} \sum_{n=1}^N \frac{1}{C} \sum_{c=1}^C y_{n,c} \log(\sigma_c(\vec{x}_n, \vec{\Theta})) \quad (2.10)$$

Cross-entropy loss is particularly effective in training neural networks because it ensures that the model assigns higher probabilities to the correct class labels.

As ANNs have become increasingly popular and more complex, the risk of overfitting has become more prominent. Overfitting is a common problem, which refers to a situation where a neural network becomes excessively complex and fits the training data too closely. As a consequence, the model's performance on new, unseen data is poor.

To mitigate the effects of overfitting, several techniques can be employed [KGC17]. Some of these techniques include L1 and L2 regularization, where a penalty term is added to the loss function that encourages smaller weights, effectively shrinking the magnitude of the weights. Another regularization technique is dropout [Sri+14], where some of the neurons are randomly dropped out (sets to zero) in the network during training, forcing the network to learn more robust features. Early stopping is another common method, where the training process is stopped before the model has fully converged on a training set, based on a performance metric on a validation set. Also, One more extended method to regularize the training is data augmentation [SK19], where new training examples are created by applying transformations to the existing data. For instance, in vision problems, images can be flipped, rotated or scaled, among other transformations. There exist many other regularization techniques, since is an open research problem nowadays, and most of them can be combined.

2.2.2 Convolutional Neural Networks

MLPs have limitations when it comes to processing images due to their architecture. This architecture treats each pixel of an image as a separate feature, which leads to an excessive number of input neurons, making the network impractical to train. That is, as we explained in the previous section, each input feature is connected to all of the neurons from the next layer, and therefore we need nm parameters for each layer, where n and m are the numbers of parameters for the previous and next layer. Fully connected layers can become computationally slow and highly *overparameterized* when dealing with medium to large-sized images. This overparameterization can potentially lead to overfitting issues, exacerbating the problem [Alz+21].

Additionally, MLPs do not consider the spatial relationships between pixels in an image, which is crucial for image understanding.

CNNs were introduced to overcome these limitations [FM82; Lec+89], which are specifically designed for processing images and have shown exceptional performance in image recognition tasks. CNNs, instead of being connected to all units from the previous layer, each neuron in the network is only connected to its neighboring units. Furthermore, a crucial aspect of this architecture is that all units share the same parameters. This means that the number of parameters is determined solely by the size of the neighborhood around each unit, also known as the receptive field, and is independent of the input data size. An illustration of a convolutional layer processing an input image of dimensions 7×7 , using a 3×3 receptive field, can be seen in Figure 2.2. In this process, each output pixel is generated by multiplying the corresponding input pixel by the weighted learned parameters.

In the case of multichannel images, such as RGB images, each output channel in a convolution operation usually takes into account all input channels from neighboring pixels. Consider an input image represented as a tensor X with dimensions (W, H, C) , where W is the width, H is the height, and C is the number of input channels. When X is convolved with a receptive field of size $\vartheta_i \vartheta_j$ and K output channels, the resulting image will have dimensions (W, H, K) , and can be expressed using the following

2. Fundamentals

in neighboring regions. Finally, the fully connected layers perform classification based on the output of the previous layers. Although, following modern trends, these fully connected layers can be replaced by more CNN layers in order to reduce the number of parameters, modifying the kernel sizes, padding and stride, to achieve the number of output neurons to be able to classify in C classes.

CNNs are powerful because they can automatically learn features that are relevant to the image classification task, reducing the need for manual feature engineering. Additionally, CNNs can leverage the hierarchical structure of images, capturing high-level features through the deeper layers of the network. These features are then used for classification, among other tasks, enabling the network to recognize objects in images with high accuracy.

In summary, MLPs have limitations when it comes to processing images due to their architecture, while CNNs are specifically designed for image recognition tasks, leveraging the hierarchical structure of images to automatically learn relevant features.

For a more comprehensive and detailed explanation of convolutions and convolutional neural networks, as well as their use in Deep Learning, we direct the reader to Dumoulin and Visin’s work [DV18] and the survey by Li et al. [Li+22b].

2.2.2.1 Object proposal methods

Object detectors are computer vision models designed to identify and locate objects within an image. They provide a bounding box around each detected object, classifying the object within it. This differs from image classification tasks that only predict a single class for an entire image. Object detection is used in numerous applications, such as self-driving cars, surveillance, and image retrieval.

Historically, object detection was addressed using a *sliding window* approach. This method involves running a fixed-size window across an image and making a prediction at each location, often at multiple scales. For instance, the Viola-Jones detector, a well-known object detection algorithm, leverages this approach for face detection [VJ01]. However, the sliding window approach can be computationally intensive, as it requires the model to make predictions for a large number of windows (in the order of 10^6

for multi-scale predictions). Object proposal algorithms were introduced to alleviate the sliding window approach's computational burden. These algorithms aimed to reduce the number of regions to process per image by proposing a limited set of candidate bounding boxes that likely contain objects. Some of these algorithms used techniques such as merging overlapping regions [San+11] or filtering out unlikely regions based on a specific score [MWY10]. However, defining a general merging or scoring procedure that works well across various object classes and contexts proved challenging.

Region Proposal Networks (RPNs) are a more recent approach that addresses these challenges and is a critical component in modern object detection models. The key innovation behind RPNs lies in their ability to efficiently propose candidate object bounding boxes or regions in an image, significantly reducing the computational load compared to previous techniques, such as the sliding window approach. RPNs use a fully convolutional network trained to generate a set of object proposals directly from an image, each with an "objectness" score indicating the likelihood of the proposed region containing an object [Ren+17]. In essence, RPNs can be considered a learned object proposal method that can efficiently predict object regions in an end-to-end manner. In contrast to the sliding window and other object proposal methods, RPNs propose regions, referred to as "anchors", with various scales and aspect ratios at each location in the image. This multi-scale, multi-aspect design allows RPNs to handle various object sizes and shapes, providing more accurate and flexible object proposals. As a result, the number of proposals generated by an RPN is significantly fewer than those generated by the sliding window approach, making the computation more efficient and manageable. Overall, by using CNNs and learned "objectness" scoring, RPNs offer an effective and efficient solution for proposing candidate object regions in an image, significantly improving the computational efficiency and performance of object detection systems.

Faster R-CNN [Ren+17] (Figure 2.3) is an influential object detection model integrating RPNs into its architecture. The Faster R-CNN model uses a two-stage process for object detection. In the first stage, the model uses a Region Proposal Network to generate a set of object proposals. The proposals are rectangular regions that likely contain objects. In the second stage, these proposals are used by a CNN to both classify the object in each

2. Fundamentals

proposal and refine the proposal's bounding box. This two-stage process allows Faster R-CNN to effectively handle the problem of scale in object detection, which is essential when dealing with images containing objects of various sizes.

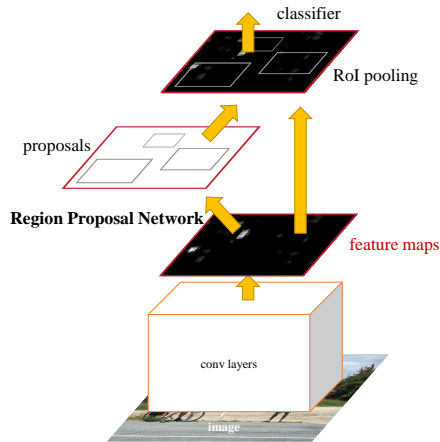


Figure 2.3: Faster R-CNN in a single, unified network for object detection. The RPN module serves as the “attention” of this unified network. Image originally taken from [Ren+17].

Mask R-CNN [He+17] extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Thus, Mask R-CNN outputs a mask (a pixel-wise segmentation of the object) for each detected object, in addition to the class label and bounding box. This enables instance segmentation, which is detecting and delineating each distinct object of interest appearing in an image. Mask R-CNN has proven to be highly effective, for instance, in segmentation tasks, providing more detailed information about an object's spatial layout and extent than the bounding box alone.

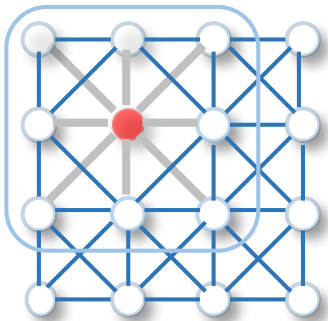
In summary, object detectors are powerful tools for locating and identifying objects within images. With the integration of Region Proposal Networks, the Faster R-CNN and Mask R-CNN models have been particularly influential, offering robust solutions for object detection and instance segmentation tasks, respectively.

2.2.3 Graph Neural Network

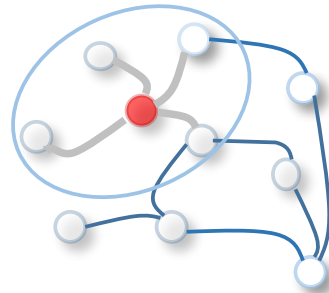
A Graph Neural Network (GNN) is a type of neural network that is specifically designed to work with graph-structured data. As explained in Section 2.1.3, a Graph \mathcal{G} is characterized by having a set of nodes or vertices, \mathcal{V} and a set of edges, \mathcal{E} , that connect those nodes.

In the previous section, we explained how CNNs work in terms of locality, where the CNN assumes that the local neighborhoods of each data point are fixed and regular, while GNNs can handle arbitrary and varying local neighborhoods based on the graph structure. In other words, CNNs are more suited for data with a fixed grid-like structure, like an image, while GNNs are more suited for graph-structured data where the neighborhood of each node can vary.

We can see a visual example in Figure 2.4. Figure 2.4a represents the pixels of an image. The neighbors of the pixel marked in red are always its surrounding 9 pixels. Figure 2.4b represents a graph where the neighbors of each node (or pixel, if it were an image) vary in each graph. In the figure, the node's neighbors marked in red are the four rounded ones.



(a) Image pixels. Analogous to a graph, each pixel in an image is taken as a node where neighbors are determined by its position. The neighbors of a node or pixel are ordered and have a fixed size.



(b) Graph. Different from image data, the neighbors of a node are unordered and variable in size.

Figure 2.4: Pixel image neighborhood vs. Graph node neighborhood. Original image taken from [Wu+19b]

2. Fundamentals

The varying locality constraint is one of the properties of GNNs. One standard way to define the neighborhood of a node v_i , \mathcal{N}_i , in an undirected graph is as follows:

$$\mathcal{N}_i = \{j \mid v_{ij} \in \mathcal{E} \vee v_{ji} \in \mathcal{E}\} \quad (2.12)$$

Thus, we can define $\mathcal{X}_{\mathcal{N}_i}$ as the multiset containing all the features of the neighborhoods:

$$\mathcal{X}_{\mathcal{N}_i} = \{\{\mathcal{X}_i \mid v_i \in \mathcal{N}_i\}\} \quad (2.13)$$

Then, we can define a local function, ϕ , which can take into account the neighborhood of a node v_i :

$$\vec{h}_i = \phi(\mathcal{X}_i, \mathcal{X}_{\mathcal{N}_i}) \quad (2.14)$$

One important topic in the graph representation is the *node ordering*. Usually, we do not have a node order, therefore the GNN local function ϕ has to be not affected by a possible permutation of the nodes and edges. So, the GNN must satisfy the invariance and equivariance rules [Vel23].

Defining ϕ is a highly active area of research in machine learning today. Depending on the context, it may be referred to as “diffusion”, “propagation”, or “message passing”. According to [Bro+21; Vel23], most of these methods can be classified into one of three spatial flavors:

$$\vec{h}_i = \phi(\mathcal{X}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij} \varphi(\mathcal{X}_j)) \quad (\text{Convolutional}) \quad (2.15)$$

$$\vec{h}_i = \phi(\mathcal{X}_i, \bigoplus_{j \in \mathcal{N}_i} a(\mathcal{X}_i, \mathcal{X}_j) \varphi(\mathcal{X}_j)) \quad (\text{Attentional}) \quad (2.16)$$

$$\vec{h}_i = \phi(\mathcal{X}_i, \bigoplus_{j \in \mathcal{N}_i} \varphi(\mathcal{X}_i, \mathcal{X}_j)) \quad (\text{Message-passing}) \quad (2.17)$$

where φ and ϕ are neural networks, such as $\varphi(x) = \sigma(Wx + b)$, from Eq. (2.5), and \bigoplus is any aggregator that is permutation-invariant, such as \sum , averaging, or max. The GNN’s expressive power increases gradually

from Eq. (2.15) to Eq. (2.17), but this may come at a cost of interpretability, scalability, or learning stability. We refer to the reader to a more extent and comprehensive guide of the convolutional GNNs, which include Chebyshev network [DBV16] and graph convolutional network [KW17], among others; a good example of representative attentional GNNs include Graph Attention Networks [Vel+17]. In this thesis, we will make use of the *Message-Passing* (MP), including graph networks [Bat+18] and Message Passing Neural Networks (MPNN) [Gil+17b].

With a GNN layer in place, we can perform various tasks on a graph by suitably combining the node features \vec{h}_i . Three primary tasks include:

Node classification: The goal here is to predict targets for each node $v_i \in \mathcal{V}$. Since the output is equivariant, we can learn a shared classifier directly on \vec{h}_i by adding a final linear layer $f(\vec{h}_i, \vec{\Theta})$, following Eq. (2.5), with c classes as output size. Some examples are the classification of header textlines [And+22] and the node classification from a conjugated graph to search for substructures in historical tables [PDM19b].

Graph classification: If the objective is to predict targets for the entire graph, we need an invariant output. This requires reducing all the \vec{h}_i into a common representation, for example, by performing $\sum_{\mathcal{V}} \vec{h}_i$, and then learning a classifier over the resulting flat vector. As we notice, it is very similar to the node classification task but with a reducing all the intermediate node embedding. An example is the document classification with graphs [YML19].

Link prediction: In this case, we might be interested in predicting properties of edges e_{ij} or even predicting the existence of an edge, which is referred to as “link prediction”. A classifier can be learned over the concatenation of features $\vec{h}_i || \vec{h}_j$, along with any given edge-level features. The “link prediction” problem can be tackled as a binary problem by classifying each edge e_{ij} in a graph. For example, in [PV21; And+22] we obtained the substructures (rows and columns) from historical tables by classifying each edge e_{ij} in a binary way to remove some edges and finally applying a connected components algorithm.

These are the most relevant tasks that can be done with GNNs, but not the only ones. For instance, graph clustering is another one, which can be seen as a binary link prediction problem, but can also be tackled in other ways. For instance, by applying a clustering algorithm after getting the

2. Fundamentals

node embeddings from an already trained GNN [Chi+19]. An example of graph clustering with GNNs is identifying communities in a social network, where each community represents a group of people with similar interests or connections [HYL17; Vel+19].

For a thorough understanding of GNNs, including tasks and their application in Deep Learning, we recommend the reader consult the extensive survey by Wu and Pan on GNNs [Wu+19b].

2.2.4 Transformer Models

Transformers are a type of neural network architecture introduced in the paper “Attention is All You Need” by Vaswani et al. [Vas+17]. They have since become a dominant architecture in Natural Language Processing (NLP), outperforming the previously prevalent recurrent neural networks (RNNs) and CNNs in many tasks.

The key innovation of the Transformer is the self-attention mechanism, which computes a weighted sum of all inputs instead of focusing only on local or sequential contexts. This gives the Transformer the ability to handle long-term dependencies in data, which is particularly useful for tasks like machine translation, text summarization, and other NLP tasks.

The Transformer architecture consists of an encoder and a decoder, each composed of a stack of identical layers. Each layer has two main sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. The self-attention mechanism, also known as scaled dot-product attention, is calculated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.18)$$

Where Q is the matrix of queries, K is the matrix of keys, V is the matrix of values, and d_k is the dimensionality of the keys, which is used for scaling. In this context, the keys and values are typically the output of an embedding layer or the output from a previous layer in the network. To allow the model to capture information from different positions, the self-attention mechanism is extended to multi-head attention. This allows the model to jointly attend to information from different positions and representation subspaces. In the multi-head attention, the queries, keys, and values are linearly projected h times with different, learned linear projections to d_k ,

d_k , and d_v dimensions, respectively. The self-attention mechanism is then applied in parallel to produce the output:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.19)$$

where each head is:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2.20)$$

Here, W^O is the output projection matrix and W_i^Q , W_i^K , W_i^V are the learned linear projections for each head.

An overview to the encoder-decoder transformer architecture can be seen in Figure 2.5. We can also see the Positional Encodings. Positional encodings are used in the Transformer architecture to give the model some information about the relative positions of words in a sentence. This is crucial because the Transformer’s self-attention mechanism has no inherent sense of position or sequence order, unlike RNNs or CNNs. Positional encodings thus inject some notion of order in the data, helping the model understand the relative importance and relationships of the words in a sentence.

Overall, the Transformer architecture’s design enables it to effectively model dependencies regardless of their distance in the input or output sequences, making it an excellent choice for many sequence-to-sequence prediction tasks. For a more comprehensive explanation of Transformers, we refer the reader to the original paper [Vas+17] and to this survey [Isl+23].

2.3 Open and Closed Set Classification

First, let’s examine the traditional Pattern Recognition (PR) classification paradigm, where each sample X in \mathcal{X} is assumed to belong to one of C known classes. This setting is referred to as “*Closed Set Classification*” (CSC). Within the minimum-error risk statistical framework, the optimal prediction of the class of X can be determined as [DH+73]:

$$c^*(X) = \arg \max_{c \in \{1, \dots, C\}} P(c | X) \quad (2.21)$$

2. Fundamentals

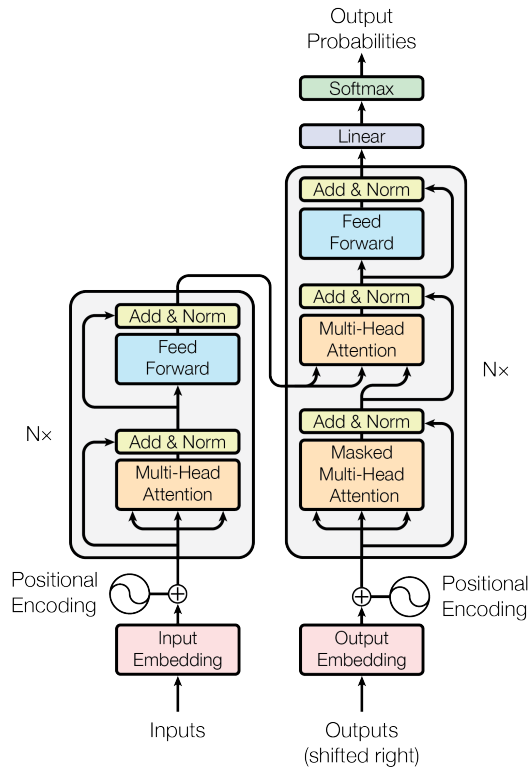


Figure 2.5: The Transformer - model architecture. Image taken from [Vas+17].

The posterior probabilities $P(c | X)$ can be calculated using various well-established methods. For instance, a common way to calculate these posteriors is using an MLP where the input is X , the output is a softmax layer with C units, and training is conducted using backpropagation with the standard cross-entropy loss. It is well known under these conditions [Bis07] that each output of the ML model, c , approximates $P(c | X)$, where $1 \leq c \leq C$. Consequently, Eq. (2.21) can be directly applied.

A CSC classifier is typically assessed by its probability of error, estimated as the *Error Rate* k_e/K , where k_e represents the number of incorrect predictions made on a test set of K image documents from the same C classes used for training [DH+73].

2.3.1 Open Set Classification

In machine learning applications, it is uncommon to have a complete set of classes during the training stage. Additionally, several classes within the available ground truth may only contain a small number of samples, making them unsuitable for training or testing. This forces the exclusion of these classes from the traditional CSC paradigm. Also, as new samples persistently emerge and require processing, the classic CSC approach proves to be inadequate. This leads to the adoption of the so-called “*Open Set Classification*” framework [GHC21; MC21; SJB14; SXL17], which assumes the existence of a larger number of potentially unknown or uncertain classes, denoted as $\tilde{C} > C$, within the sample space \mathcal{X} .

Initially, consider a configuration where the system may be trained using samples from all C *known classes*, in addition to an *extra* “REJECT class” that encompasses the residual $\tilde{C} - C$ unknown classes. All the GT classes with insufficient samples can be suitably incorporated into this “class”. This remains a relatively conventional Pattern Recognition (PR) setting, which entails training and classification with $C' = C + 1$ classes [DH+73]. The minimum error-risk classification is provided by Eq. (2.21), replacing C with C' , and the traditional “*Error Rate*” can still be reasonably employed for Open Set Classification (OSC) evaluation.

One advantage of using the REJECT class is to learn the distribution data from that class, useful if we know beforehand that the new data incoming will be similar to these REJECTS. Another advantage is that we do not need to decide whether to reject or not a sample, so we do not need a threshold. One disadvantage is we need data to train that class, and in a real case may not have this data.

An alternative approach for handling test samples from *unknown classes* involves training the system exclusively with samples from the C *known classes*. Subsequently, a threshold t must be established, which indicates a class posterior probability below which any test sample should be rejected, meaning it is considered to belong to a REJECT class. Formally, let $Q(X) \stackrel{\text{def}}{=} \max_{1 \leq c \leq C} P(c | X)$. Then:

$$c^*(X) = \begin{cases} \arg \max_{c \in \{1, \dots, C\}} P(c | X) & \text{if } Q(X) \geq t \\ \text{REJECT} & \text{otherwise} \end{cases} \quad (2.22)$$

2. Fundamentals

Under this scheme, various methods can be employed for OSC with REJECT and training solely involving the C known classes.

Several OSC approaches have been introduced in recent literature. For instance, the model proposed in [SXL17], referred to as “one versus rest” (1-vs-rest), configures the output layer of a neural network as a vector of C *sigmoid* activation functions. In this configuration, each output c corresponds to a Bernoulli distribution, $P(b_c | X)$, $1 \leq c \leq C$, where b_c represents the value of a binary random variable that is 1 if the class of X is c and 0 otherwise.

Alternatively, in [Yan+22a], a Convolutional Prototype Network (CPN) is presented as a versatile approach for both OSC and CSC. In this work, an input convolutional stack is dedicated to extracting features from the input, typically images.

If a single, fixed threshold t can be assumed or estimated, ML models can effortlessly implement OSC with REJECT as in Eq. (2.22). This is achieved by considering $P(c | X)$, $1 \leq c \leq C$, as the output probabilities provided, for instance, by an MLP.

Allowing the user to adjust the reject threshold is a practical option that enables tailoring a trained system to the rejection requirements of each specific batch of data. To evaluate rejection performance in this context, a ROC curve [MRS08] can be plotted, characterizing the system for all possible thresholds. The area under this curve, known as AUROC, serves as a widely accepted scalar measure that adequately assesses the system’s overall performance across all reject thresholds. A ROC curve is based on binary decisions, such as determining whether a sample belongs to one of the C known classes.

Document Information Extraction And Classification Overview

In this chapter, we explore the significance of document layout analysis (DLA) and document classification (DC) in the context of historical manuscripts from an information extraction point of view.

Information extraction (IE) in documents refers to the process of automatically identifying and extracting relevant information, such as entities, relationships, or specific data points from documents. This process typically involves the use of natural language processing (NLP), machine learning (ML), and other computational techniques to analyze and interpret the content of documents, transforming them into structured data that can be easily processed, analyzed, and stored. The objective of IE is to make data processable, a fundamental aspect of language processing. While automatic IE has been around as long as document databases, its technology has matured in the past decade. This evolution reflects the broader journey of artificial intelligence. This journey has seen three main phases: the rule and dictionary-based approach, the statistical machine learning approach, and the deep learning approach[Yan+22b], as we outline in a further section.

The case of interest in this thesis, historical documents, holds particular intrigue due to the challenges presented by different corpora. For instance, in the case of handwritten tables, there is sometimes a printed layout that aims to delineate each cell of the tables. At other times, these layouts are not printed but are merely hand-drawn lines attempting to serve the same purpose. Even sometimes, there might not be a physical layout but a logical one, without any delimitating lines in the layout.

These difficulties mean that at first glance, there is no apparent separation between cells or regions, but upon closer reading of the text, one would discern it. However, even in these scenarios, there are instances where the text is hard to read even for experts. The original conditions under which these pages were written were not always ideal, compounded by the wear

3. Document Information Extraction And Classification Overview

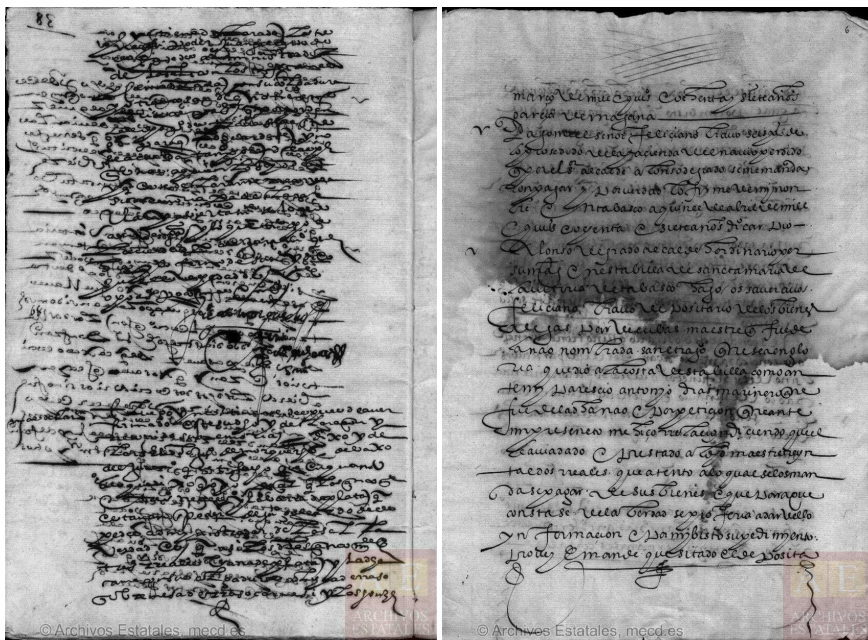


Figure 3.1: Two examples from the Archivo General de Indias. Severe degradation can be observed

and tear over time. For example, there are documents penned aboard a ship, where writing was done hastily, and sometimes, the ink would spread due to water splashes. In Figure 3.1, we can see two examples of this. The image on the left shows significant bleed-through and considerable wear, including smudged ink. The image on the right illustrates how the book's page got wet, further degrading its quality.

On the other hand, Document Layout Analysis (DLA) addresses the challenge of identifying the inherent structure of documents. Its objective is to extract all structural elements of a document and the possible relationships between them. For example, DLA seeks to understand the distribution of information, usually across a document's pages: the location of text lines, how they are organized into groups (such as paragraphs), the presence of illustrations and their relationship to the surrounding text, and so on.

The layout plays a vital role in improving the accuracy and efficiency of the process. Understanding the layout can provide valuable context and guide the identification of relevant sections or data points within the

document. By incorporating layout analysis in conjunction with other IE techniques, we can better comprehend the organization and structure of the content, ultimately enhancing the extraction process and unlocking deeper insights into historical documents.

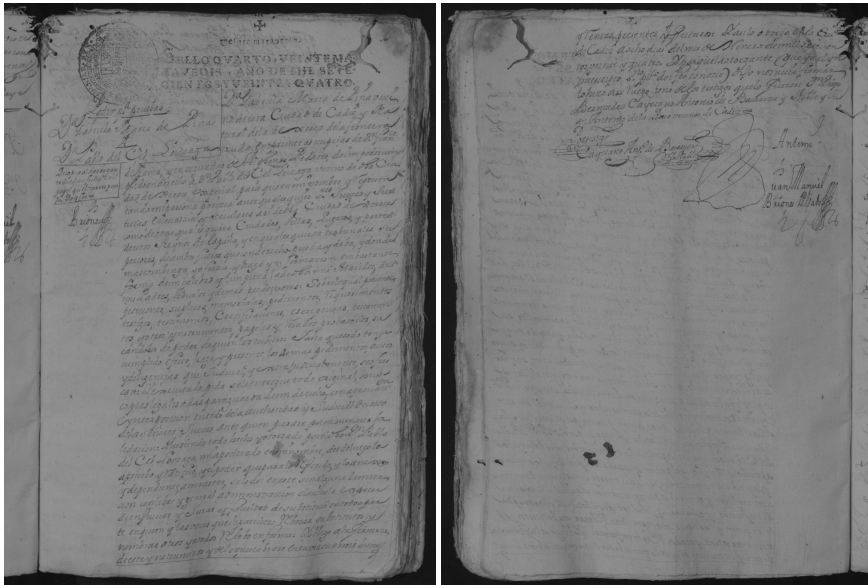
The layout carries not only vital information but also provides insights into the cultural and historical context. This chapter delves into the challenges associated with obtaining layouts of historical documents, which can be particularly difficult to analyze due to their poor condition and deterioration over time. The usefulness of the layout or textual information, or perhaps both, depends on the specific task at hand and the characteristics and deterioration of the documents in the dataset.

To illustrate this point, let's consider the deed segmentation and classification in the Provincial Historical Archive of Cádiz (AHPC, by its Spanish acronym) dataset. Figure 3.2 shows two images that together form a notarial deed from the JMBD4950 folder in the dataset of the AHPC dataset. These two images constitute a single document, which must be assigned a class based on the type of notarial deed it represents. The layout of the images can greatly help us to determine that the image on the left, Figure 3.2 (a), is a page that initiates the notarial deed. This is a common pattern in this dataset, as these deeds have certain visual characteristics that aid in segmenting pages that make up a notarial deed. For instance, deeds always begin on left pages, often have a text box with a slightly different layout from the main body, and sometimes feature a stamp, among other attributes.

On the other hand, at the end of the deed, as seen in the page of Figure 3.2 (b), it usually contains a signature, the text might not fill the full page, and they tend to conclude on the right pages, among other features. While these characteristics do not always hold, they are quite common. As such, the layout of the images is very helpful for determining the beginning and end of a notarial deed within this dataset.

However, identifying the class of these notarial deeds becomes much more challenging based on the image alone. Typically, we need to know the textual content of the images to assign a class. Although we can gain some clues from the images themselves (especially if we look at the text box on the pages that start the deed), the poor preservation of some images makes it impossible to see the textual content. This is where the textual content comes into play; in addition to providing richer information about

3. Document Information Extraction And Classification Overview



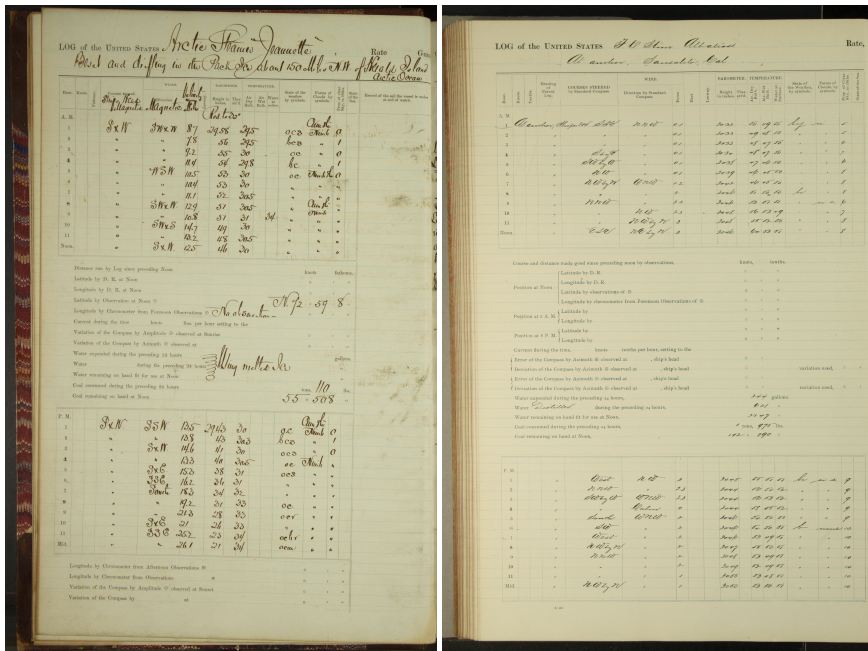
(a) 72th page of the JMBD4950 book. (b) 73th page of the JMBD4950 book .

Figure 3.2: A notarial deed composed by two pages from the JMBD4950 book, AHPC dataset.

the content of the deed itself, it is easier to process text from different pages within the same system than if they were images, as we will see in Section 4.4.

In addition to the challenges presented by historical document classification and segmentation, extracting information from structured documents, such as tables, provides another clear example of how knowing the document’s class or at least the type of layout (which could be considered a class) can help develop ad-hoc methods. In this case, let’s consider two examples from the same dataset: Jeannette and Albatross. These documents, while similar due to their common origin, have clear differences, mainly in the shape of the tables and their headers, which complicates the extraction task.

In Figure 3.3, we can see the Jeannette and Albatross images side by side. These images show the differences in the table layouts, particularly in the headers, which are crucial for understanding the table’s content and extracting information from it. Being able to identify the type or class of layout that each table belongs to would be incredibly helpful in extracting



(a) A left page from the Jeannette book. (b) A left page from the Albatross book.
 Figure 3.3: Two left pages that belong to two different books, from the Hisclima dataset.

information from structured documents like these.

Despite these difficulties, understanding the layouts and textual content in historical manuscripts remains crucial for successful IE and DC tasks, ultimately enabling more accurate and comprehensive analysis of these invaluable resources.

In summary, the analysis of historical manuscript layouts poses unique challenges due to several factors:

Age and deterioration: Over time, historical documents, especially those centuries old, tend to deteriorate significantly. This can lead to faded text, damaged pages, and other artifacts, complicating layout analysis and other tasks.

Variability in writing styles and conventions: Writing styles and conventions have evolved considerably throughout history. As a result,

3. Document Information Extraction And Classification Overview

historical manuscripts may exhibit layouts that are inconsistent with or unfamiliar to modern standards, making it difficult to identify and understand the content's organization.

Complex layouts: Some historical documents may feature intricate and elaborate layouts, incorporating a mix of text, images, and decorative elements. This complexity can make more difficult the identification and segmentation of different layout components.

Dataset: In addition to all these challenges, there is typically not a lot of data in each dataset. This contrasts with recent trends in the field of deep learning, where datasets are becoming larger, and architectures are often designed to leverage vast amounts of data. In the case of historical archives, not only are they considerably different from one another due to many of the factors explained, but labeling them is also much more expensive since it requires highly skilled personnel to do so.

3.1 Document Layout: Transcending Single Pages

A document's layout goes beyond the organization of elements on a single physical page. It refers to the overall arrangement and structure of the entire document, which can range from less than a page to hundreds of pages, or even an entire book. The layout comprises various components such as text blocks, images, tables, headings, and margins, as well as the relationships among these elements.

A comprehensive understanding of a document's layout is crucial for tasks like IE and DC. For information extraction, the layout offers valuable context and assists in identifying relevant sections or data points within the document. In DC, the layout serves as a critical feature for differentiating various document types or comprehending the structure and organization of the content.

In this thesis, we consider the challenge of obtaining the structure of an entire book, which involves identifying the semantic units that compose it, such as chapters or notarial acts, and the containing text. The ultimate goal is to separate the text of each semantic unit from the rest, allowing for future classification or information extraction from each unit individually.

Formally, we aim to obtain the sequence of semantic units $D = D^1, \dots, D^K$, where each D^k is a semantic unit, and K is the total number of semantic units in the book.

Additionally, we can represent the book-level segmentation as a solution to an optimization problem, where we aim to find the most probable segmentation h that explains the intrinsic structure of the book B . Under the Maximum a Posteriori Probability framework, this can be expressed as:

$$\hat{h} = \arg \max_{h \in \mathcal{H}} P(h | B) \quad (3.1)$$

where \mathcal{H} is the set of all possible book segmentations. Moreover, each segmentation h includes both geometric and logical information, and h can be represented in various ways (such as a graph, a tree, or a list of elements).

Earlier, we saw a notarial act consisting of two consecutive pages from the JMBD4950 folder in Figure 3.2. In Figure 3.4, we can see another act from the same folder, but in this case, the act consists of six consecutive pages. In this corpus, the acts always start on a new page, so we will not encounter a notarial act that merges with another one on the same page. This, as we will see later, allows us to try to solve the problem in a slightly different and more direct way, starting with image-by-image classification and subsequently processing book-by-book.

However, notarial acts in other collections do not always start on separate pages. It is common to find them in a mixture of text where it is usually necessary to segment the page into different pieces and combine them with others on other pages of the corpus to complete the information of the act. This means that page-level classification is not sufficient; we need more refined systems that work at the line level or even at the pixel level, to subsequently combine everything and work at the book level.

In Figure 3.5, we can see an example of an act in the Nesle corpus. Each text piece is labeled in the image according to whether it is a beginning (I), middle (M), end (F) of an act, or a complete act (C), starting and ending in the same paragraph. This notation is explained in detail in Chapter 4. We can see how the first act spans almost four pages, spread across two images. If one wanted to extract information or classify this act, it would first be necessary to segment it as shown in the image, and then transcribe and/or search for the information.

3. Document Information Extraction And Classification Overview

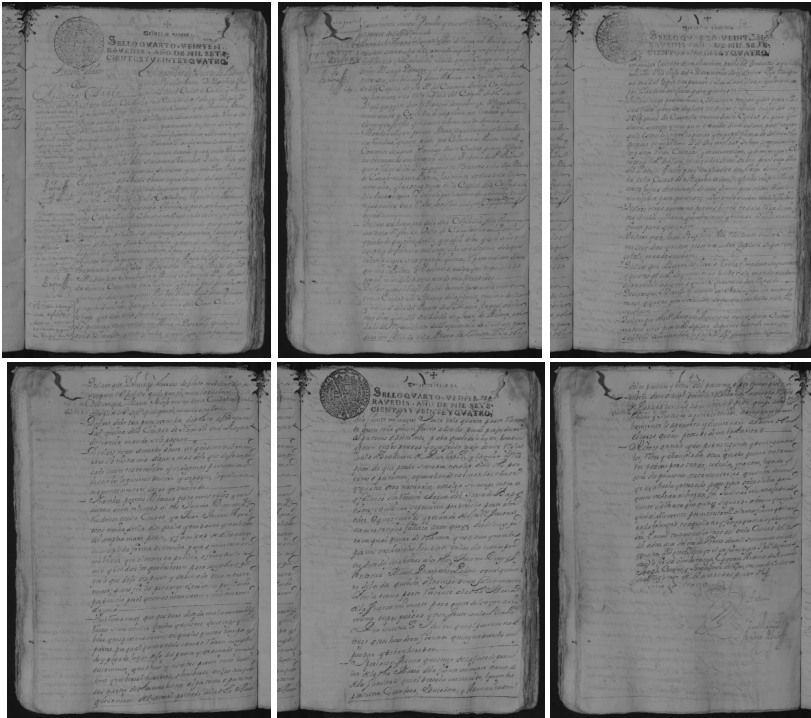


Figure 3.4: Example of an act of six consecutive pages from the JMBD4950 corpus from AHPC.

In Figure 3.6, we show another example of two double-column pages from the Denis corpus. In these images, we see a total of nine complete acts and two more incomplete ones. In the first image, there is an act that began on another page but ended on this one. With the “C” label, we see a total of eight acts, which start and end in the same paragraph. Between the two pages, we see an act that begins on one and ends on the next, labeled with the sequence “IF”.

These examples demonstrate the need to work beyond the page level to extract this information.

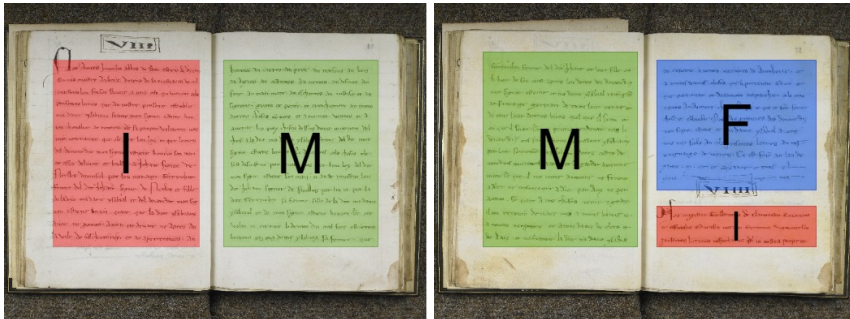


Figure 3.5: Example of two consecutive pages from the Nesle corpus. Two acts are shown, creating the sequence “IMMFI” for the first and “I” for an unfinished second act, which will finish in the following pages.

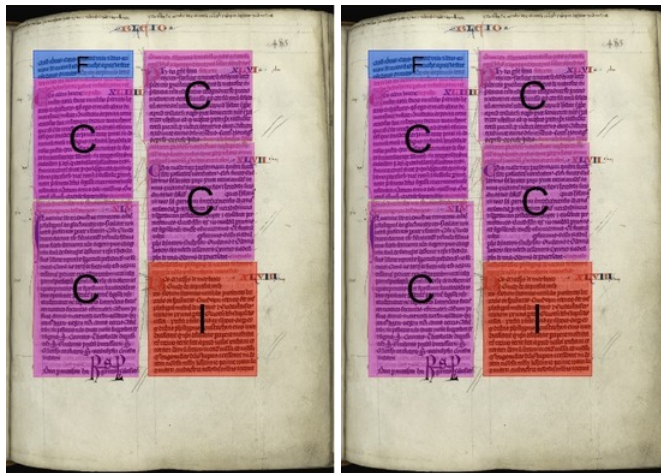


Figure 3.6: Example of two consecutive pages from the Denis corpus. There are a total of nine acts in nine different blocks, creating the sequence “FCCCCIFCCCCI”. The first one, “F”, is finishing a previous act, then we have nine acts “CCCC”, “IF” and “CCCC”, and a last and unfinished act “I”, which will finish in the following pages.

3. Document Information Extraction And Classification Overview

3.1.1 Related Works

There are several ways to approach this problem. One method could be transcribing the entire book, obtaining its reading order, and then segmenting it into semantic units based solely on the transcription. Depending on the dataset, this approach may work for non-historical text, as there would be fewer transcription errors, and a more or less consistent pattern could be followed to separate each unit. However, in historical documents, this approach is often impractical due to transcription errors that arise.

Another approach could be based on the visual layout of each image, searching for specific patterns or rules that apply to each book. In the field of Table of Contents (ToC) extraction, this is typically achieved by creating a set of rules to obtain the ToC of digitized books. This task involves obtaining a triplet for each book chapter: the title, the starting page, and the depth level of that chapter.

In recent years, various competitions have been held on the extraction of ToC [Dou+09; DKM11; WMG13], and other authors have continued to improve systems for this extraction [NDC17]. Although ToC extraction is a related task that appears close to the segmentation and extraction of notarial acts in historical books, segmentation still presents challenges not encountered in ToC extraction in modern books.

Working with historical documents presents a unique set of challenges and difficulties not found in non-manuscript documents, as mentioned in the introduction. These challenges make it impossible for a set of ad-hoc rules to function effectively. In this thesis, we will work with the image, or the image and text, to obtain the semantic units of a book and segment it accordingly.

3.2 Document Classification of Historical Manuscripts

As mentioned previously, document classification, i.e., assigning a class or “type” to each document, can help us extract information from them. In this thesis, we also focus on classifying not just single pages but documents consisting of multiple pages. From now on, bundles, boxes, books, or folders of manuscript images will be called “image bundles” or simply

“bundles”. A bundle may contain several, often many, “image documents”, also known as “records”, “acts”, – or “deeds” in the case of notarial image documents. Image documents are assumed to belong to “types” or classes, perhaps the most crucial information needed to describe a manuscript.

In DC tasks, the layout can provide some insights into the organization and structure of the content, which can help identify specific document genres or subcategories. For example, legal documents, scientific articles, or religious texts may exhibit unique layout characteristics useful for classification. However, historical documents might not be well-preserved due to the deterioration of the paper, which can make it difficult to rely on layout features for classification.

Thus, the task we are interested in is classifying handwritten image documents, which can range from a few to dozens or even hundreds of handwritten text images, into a set of classes or types associated with the topics or content (semantics) conveyed by the written text in the images. We refer to this task as “content-based image document classification” (CBIDC).

In such cases, the textual content of the documents is typically more significant for distinguishing between various document types or historical periods. Although the textual content might also be challenging to obtain from historical manuscripts in poor condition, techniques like probabilistic indexing can be particularly helpful for extracting meaningful information from these documents.

For example, the document shown in Figure 3.4 belongs to the “will” class. We can primarily determine this because, on the first page of the act, it is clearly stated, as seen in Figure 3.7.

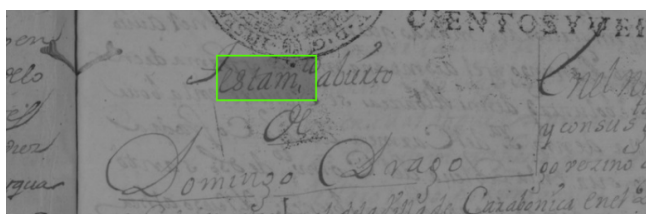
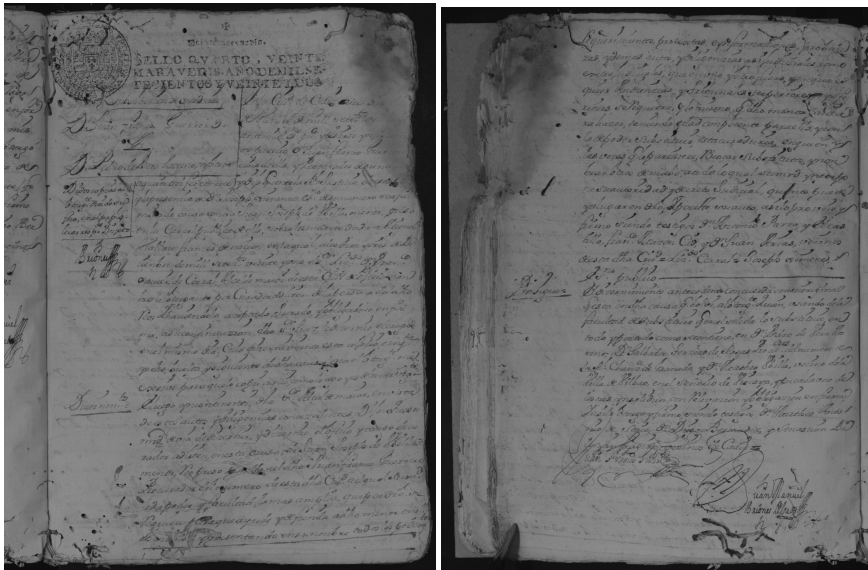


Figure 3.7: Page 78 from the folder JMBD4950. The abbreviation of the word “Testamento” (“Will” in Spanish) is clearly stated as “Testam.”.

However, other documents are not as easily classified due to degradation or ambiguous content that could even challenge an expert’s judgment.

3. Document Information Extraction And Classification Overview

Sometimes, both issues coincide. In Figure 3.8, we see another document where the assigned class would be “Power of Attorney”. Still, the keywords for this type of document (which are typically “Poder”, meaning “Power of Attorney” in Spanish, or “Substitución” (“Substitution”)) are neither located in the same position nor on the first page.



(a) 72th page of the JMBD4950 book. (b) 73th page of the JMBD4950 book .

Figure 3.8: A notarial deed of class “Power of Attorney”, composed by two pages from the JMBD4946 book, AHPC dataset.

Given these challenges, it is clear that the first page only sometimes is helpful, and we need to rely on the rest of the document’s textual content for classification. Due to the poor condition of the documents and uncertain transcriptions, classifying these documents poses a significant challenge.

3.2.1 Probabilistic Indexing of Handwritten Text Images

The Probabilistic Indexing (PrIx) framework was developed to address the inherent word-level uncertainty typically found in handwritten text images, particularly in historical manuscripts. Within this framework, any image element highly likely to be interpreted as a word is identified and

stored, along with its *relevance probability* (RP) and location in the image. These text elements are referred to as “*pseudo-word spots*”.

Following [Tos+16; Pui18], the RP for an image-region x and a pseudo-word v is denoted as $P(R = 1 \mid X = x, V = v)$, but for conciseness, the random variable names will be omitted, and for $R = 1$, we will simply write R . As discussed in [VTP21], this RP can be approximated as:

$$P(R \mid x, v) = \sum_{b \sqsubseteq x} P(R, b \mid x, v) \approx \max_{b \sqsubseteq x} P(v \mid x, b) \quad (3.2)$$

where b represents a small, word-sized image sub-region or Bounding Box (BB), and $b \sqsubseteq x$ denotes the set of all BBs contained in x . Note that $P(v \mid x, b)$ is the posterior probability needed to “recognize” the BB image (x, b) . Thus, assuming the computational complexity entailed by (3.2) is algorithmically managed [Pui18], any sufficiently accurate isolated word classifier can be used to obtain $P(R \mid x, v)$. In this case, we employ the methods described in [Pui18]. This word-level indexing approach has proven to be highly robust and has been successfully used to index several large iconic manuscript collections, such as the French Chancery collection [Blu+17], the Bentham papers [Tos+19], and the Spanish Carabela collection [Vid+20].

In summary, while the layout can serve as an additional feature in DC, the primary focus should be on the textual content, especially when working with historical documents in poor condition. By incorporating advanced techniques like probabilistic indexing, we can overcome the challenges associated with analyzing historical manuscripts and improve classification performance.

3.2.2 Related Works

Existing approaches for content-based DC typically assume that documents consist of electronic text, with characters, words, and paragraphs unambiguously given. Therefore, the conventional method to address the proposed CBIDC task would be transcribing the images and then applying off-the-shelf DC techniques. However, manual transcription is not feasible, and achieving accurate automatic transcripts is generally unattainable or unreachable for large sets of historical manuscripts. Word recognition accuracies for HTR in historical manuscripts similar to those examined in this

3. Document Information Extraction And Classification Overview

thesis have been reported to be as high as 40-60% in [Sán+19; Rom+19b; Vid+20].

It is essential to distinguish the CBIDC task from other related tasks with similar names. For example, “document classification” (DC, mentioned above, which only applies to unambiguous electronic text), “content-based image classification” (applied to single pictures of natural scenes – not text), or “document image classification” (where classes are associated with the visual appearance or page layout of single images).

First and foremost, this task is distinct from what the computer vision and image analysis literature usually refers to as “image classification” [PSM10; Lin+11; RW17], where images are classified based on global features related to colors, textures, shapes, etc. It also differs significantly from the task often called “content-based image classification” [PLK04; KKJ12]. In a conventional content-based image classification task, images typically contain large objects, such as mountains, animals, vehicles, or persons, out of a few tens (or maybe a few thousand) object types. In contrast, a typical text image contains several hundred small and detailed “objects” (i.e., words) out of tens or hundreds of thousands of different “object types” (i.e., different words in a natural language lexicon). For similar reasons, works such as [Pae+99; TZZ13], combining visual and text features, differ from the work presented in this thesis.

Another confusion worth avoiding is relating the task considered here with what is often called “image document classification” in the document analysis literature, where images of printed or handwritten text are classified based on more or less global features such as layout, visual shape, type of script, writer (hand), etc. [CB07; Kan+14; XLW17].

It is important to note that recent works on document classification, such as those involving multimodal approaches and visual transformers [SOE22; Xu+21], are not directly applicable to our CBIDC task. The nature and size of the textual visual objects in our task (potentially hundreds of page images) are significantly different and considerably larger than the single-image objects considered in these studies.

Instead, our objective is akin to the well-established and widely recognized task of content-based document classification, which assumes the data consists of plain text rather than handwritten text image documents. Classic examples for which popular datasets are available include *Twenty*

News Groups, Reuters, WebKB, etc. [MRS08; Kha+10; AZ12]

It is crucial to recognize that document types evolve over time, and in a realistic scenario, we must handle image documents of classes that have never been encountered before. In the traditional classification framework, these new image documents would be consistently misclassified. Thus, to adequately address the proposed task, new image documents that do not belong to any known class should be detected; in other words, the system should refuse or “*reject*” their classification. A key contribution of this thesis in terms of DC is to explicitly tackle this comprehensive CBIDC problem and provide satisfactory solutions within the so-called *Open Set Classification*” (OSC) framework [SJB14; GHC21; MC21].

OSC has been explored in several recent studies, such as [Yos+19; Hua+22; CG22; Shu+19; Yan+22a; SXL17]. Although most approaches proposed in these works are not directly applicable to our CBIDC task, we have successfully adapted ideas from [Yan+22a; SXL17] and compared the resulting methods with the other approaches we propose.

3.3 Information Extraction in Historical Manuscripts

One of the most challenging aspects of information extraction from historical documents is the analysis and extraction of information from historical tables. Tables in historical documents exhibit a wide variety of formats, with different alignments of rows and columns and unconventional typography. Given this flexibility in the layout considered in the past [Lan+18], performing information extraction from historical tables is a complex and challenging problem. This issue is sometimes also found in more recent electronic documents [Wei+21]. Moreover, in historical documents, we encounter the deterioration of the document over the years.

Since there are many large collections in which information was recorded in a tabular form, the interest in information extraction from such documents is immense. Some examples of documents include border records, military records, hospital records, records related to industrial processes, financial records, population records, forestry records, and travel records, among others. Extracting relevant information from these collections would enable researchers to study the past more thoroughly.

3. Document Information Extraction And Classification Overview

Two current limitations for obtaining valuable results in these collections are, on the one hand, the complex tabular layout they present and, on the other hand, the current HTR results, which are not error-free. The tabular layout can be very flexible in some situations, with lines of text running from left to right in the image [Rom+13]. In this case, the line extraction process can be very successful, and, as a result, the HTR results and relevant information extraction can be good, as they can rely on a helpful context [Rom+19a]. The tabular layout can also be composed of pre-printed sheets [RS20]. It has been shown that extracting row lines from side to side of the page image is not very helpful because the linguistic context along the columns of the same row may not be useful [RS20], and it seems better to extract lines at the cell level. In such a case, line detection can be challenging since sometimes only quotation marks are written to indicate that the value of the previous row is repeated. Furthermore, the HTR results and relevant information extraction may not be very good, as they cannot rely on the HTR context [RS20].

In this thesis, the task at hand is extracting information in the format of triplets composed of three values: a column header, a time of day (row header), and the content. The column header and the time of day form the “query”, while the content forms the “value”. The primary challenge we encounter in the datasets we use are those mentioned earlier, with different layouts and writing styles. However, the framework presented in Section 6.1, although we demonstrate the efficacy in specific cases using these triplets, is easily adaptable to any type of table or even form. Nevertheless, we believe that these datasets and queries present a significant challenge and are useful in demonstrating the system’s effectiveness.

The challenges we aim to address when performing IE on historical handwritten tables in this thesis include working with various layouts and calligraphies throughout the structured documents, as seen in Figure 3.3. Other challenges are multi-span cells, where column headers comprise more than one cell. In Figure 3.9 a), two examples can be seen where the queries for that column would be “*Clouds forms of by symbols*”, “*Clouds Moving From*”, and “*Clouds Am’t scale 0 to 10*”. Additionally, in each example, the layout changes in some cells from horizontal to vertical. In Figure 3.9 b), we see cell size differences. Another significant challenge, and one of the main reasons why a template for each layout could not be used, is seen

in Figure 3.9 c), where red lines mark the physical division of that column (left image) or row (right image), but it is observed that the cell occupies up to 3 rows or columns, without respecting the layout. In Figure 3.9 d), e), and f), we observe how printed and handwritten text are mixed in the same cell, as well as crossed-out words and the use of quotation marks to refer to others.

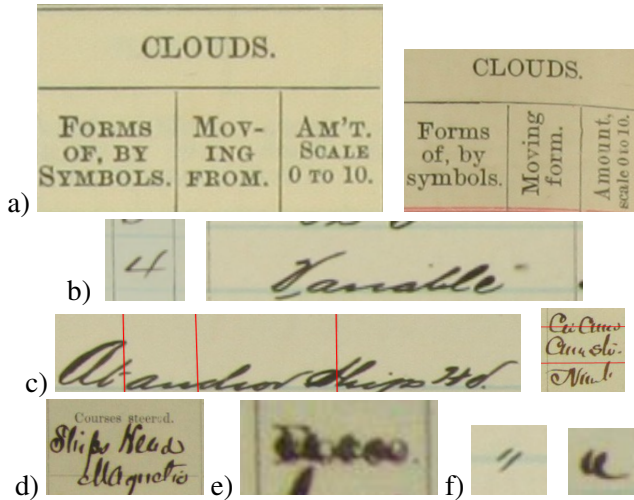


Figure 3.9: Some examples of the challenges encountered in the Jeannette and Albatross datasets are as follows: In a), we see column headers with multi-span cells above them (both), attributes written differently depending on the table layout (both), and vertically-oriented text (right). In b), we observe the differences in width between two cells. In c), we see two examples of cell contents that do not respect cell boundaries (exceeded boundaries are denoted in red). In d), we find a column header with part of its contents handwritten, and in e), a crossed-out column header is visible. Finally, in f), we observe some examples of quotation marks.

3.3.1 Related Works

In recent years, numerous studies have focused on information extraction (IE) in structured documents; however, most assume working with digital and/or non-manuscript documents.

3. Document Information Extraction And Classification Overview

In [Yan+22b], Yang et al. outline three historical divisions in the development of IE. The first methods relied on rules and dictionaries. Rule-based methods utilize extensive sets of rules and templates crafted by experts to extract information. On the other hand, dictionary-based methods searched dictionaries or domain knowledge databases to identify and extract information. Both methods typically worked on smaller datasets but demanded significant time and expertise. Typically, these methods were language-dependent, making them challenging to adapt to other languages.

Next, we have methods based on statistical machine learning. These relied on supervised training from properly labeled datasets and were used to train machine learning models like HMM, or SVM. These methods considerably improved results but required vast amounts of data for training. Additionally, there was often a need for moderately complex data preprocessing (feature engineering) by experts to harness the capabilities of ML models. Thus, both language experts and ML technique experts were needed.

Lastly, we arrive at techniques based on deep learning. This most recent approach aims to solve the problem by automatically identifying complex patterns without the need for extensive feature engineering. These are typically methods suitable for large datasets and are often capable of generalizing and detecting hard-to-spot patterns. Common methods include the use of CNNs, RNNs, MLPs, and transformers.

For instance, Gilani et al. [Gil+17a] and Siddiqui et al. [Sid+19] employed convolutional neural networks (CNNs) for detecting tables, and Siddiqui et al. [Sid+19] even attempted to recognize their structure, although they did not perform IE. Adiga et al. [Adi+19] approached the problem differently, scanning the document with OCR software and then classifying relationships between each detected word using a multilayer perceptron (MLP). Identifying relationships between entities, such as words or lines, has gained increasing importance in the community, prompting a natural transition toward working with GNNs. Similar to Adiga et al. [Adi+19], Riba et al. [Rib+19] classified relationships between previously detected words using a GNN. In contrast, Qasim et al. [QMS19] did not rely on words detected by OCR; instead, they used a CNN to extract visual features while simultaneously detecting structures like rows, columns, and cells. Subsequently, they employed a GNN to identify relationships between these

structures.

However, all these methods assume high-quality and regular images and have not been applied to historical handwritten documents.

With historical handwritten documents, we face numerous challenges previously explained that render most advanced techniques inapplicable.

Nonetheless, progress has been made in the field of historical structured documents. In the B track of the competition “*Table Detection and Recognition (cTDaR)*” [Gao+19], the detection of the structure of handwritten tables was addressed, with one team constructing an adjacency matrix based on objects detected by a CNN. Dejean et al. [DM19] attempted to create virtual objects and classify them to detect rows in tables. Later, Prasad et al. [PDM19a] improved their work by utilizing GNNs. They first created an initial graph from text lines and then pruned it by classifying the edges of the graph to find substructures, such as rows and columns. This edge classification for pruning was performed by classifying nodes on the initial conjugate graph¹.

The mentioned works focused their efforts on recognizing structures in structured documents but did not perform information extraction (IE). Some studies have been conducted to perform IE on historical tables. Lang et al. [Lan+18] and Romero et al. [RS21] accomplished this by using specific geometry-based heuristics and prior knowledge of column headers. However, the results were only presented using ground truth lines and not those automatically detected.

Additionally, Constum et al. [Con+22] presented a complete pipeline for extracting information on historical handwritten tables. However, they relied on a corpus without variations in layout, which was highly homogeneous and lacked the difficulties mentioned earlier. This made it possible to create a language model at the column level and process each row as a single line, with regularly segmented rows. Unfortunately, the results for information extraction were not reported.

¹In a conjugate graph, broadly speaking, edges become nodes and vice versa

Act Segmentation and Layout Analysis

4

Archives around the world have vast collections of historical manuscripts containing crucial notarial documents. Many of these collections have been transformed into high-resolution digital images. These manuscript images are typically stacked sequentially and organized into folders, books, or boxes. We often refer to these sequences of digitized documents as “image bundles”, where each can contain thousands of images and hundreds of records. Henceforth, these containers, whether they are folders, books, or boxes, will be termed “*bundles*”. Inside a bundle, there are multiple “image documents”, also known as “files”, “acts”, or specifically for notarial purposes, “*deeds*”. In this thesis, we use the terms “deeds” and “acts” interchangeably, both referring to the same concept. These acts, often spanning multiple pages, represent a series of semantic text segments that align logically with other elements in the pages or books, like marginalia, headers, or other text sections. An example is notarial records, in which a king would issue orders, declare inheritances, or grant powers of attorney, among other things.

These acts usually contain crucial information, emphasizing the importance of information extraction. Given the vastness of these document series, archives frequently struggle to provide detailed metadata that captures the content of each bundle. Information about the location of each act within the multitude of digitized documents is usually unavailable. It is essential to have automated solutions to assist specialists in cataloging these extensive series. Current HTR systems typically process documents on a page-by-page basis, often lacking the contextual understanding needed to effectively segment these acts. Of course, this is true for any type of book, newspaper, etc. HTR systems are usually based on line-level results [Cam20; Qui22], paragraph-level results [Blu16; BLM16; CCP22], or even page-level results [CCP23; Kim+22]. LA systems have also been

4. Act Segmentation and Layout Analysis

applied at the page level so far [BKP22; OSK18; BKP20; QTV19; Bis+21; Qui22]. Given that previous work has addressed up to the page level, one of the first steps in this task is segmenting the bundles into their individual acts, going beyond the page level, as acts can be spread across one or several pages.

As introduced earlier in Section 3.1, $D = D^1, \dots, D^K$ represents the sequence of semantic units we aim to extract, where K denotes the total number of semantic units in each bundle or book, which may vary among them.

The book-level segmentation can be depicted as a solution to an optimization problem, where our goal is to determine the most probable segmentation, h , that accounts for the intrinsic structure of a collection of images from a bundle, $B = G_1, \dots, G_M$. Within the Maximum Posteriori Probability framework, this can be formulated following Eq. (3.1), where H represents the set of all possible book segmentations. Furthermore, each segmentation h encompasses both geometric and logical information, and h can be expressed in various forms (such as a graph, a tree, or a list of elements).

In the following, we divide the problem into two different problems.

On the one hand, we assume that the physical separation between acts is limited to one page. With this simplification, we first attempt to resolve the book segmentation into notarial acts.

On the other hand, later on, we not make any assumptions or simplifications regarding the beginning and end of each act. Although the problem remains the same, i.e., segmenting books into acts or chapters, and part of the proposed solution is indeed shared, the approach and methods employed with the first assumption are more straightforward, making the separation worthwhile.

Finally, to ascertain the robustness and reliability of the architecture, we conduct tests for both challenges on a single corpus.

4.1 Problem Definition

We define a book B as a sequence $G_1 \dots G_N$ of contiguous and ordered pages typically containing text or relevant information. This book, in turn, forms a sequence of K notarial acts, $D = D^1, \dots, D^K$, where each

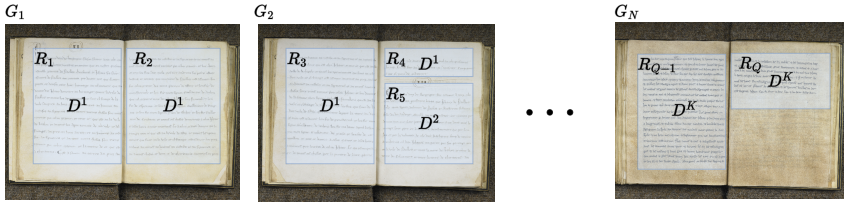


Figure 4.1: Example of notation of acts. We depict the tagging of images, regions and acts on three different pages.

element D^k is a notarial act. These acts, usually distributed across several pages, comprise a series of semantic text units that follow a logical sequence relative to other elements on the pages or books, such as marginalia, headers, or other paragraphs of text. An example would be notarial records, in which a king issued orders, declared inheritances, or granted powers, among other things.

Each notarial act D^k , $1 \leq k \leq K$, is also defined as a sequence of regions R_1, \dots, R_Q , where each region R_q , $1 \leq q \leq Q$, is defined by its text and encapsulate it. Then, each D^k act has $M(k) \leq Q$ number of regions R_q . This text contains the information we want to extract from each act and separate it from the rest of the notarial acts.

While each region R_q that makes up an act is defined and delimited by the text it contains, it also has a geometry $\vec{r} \in \mathbb{R}^4$ that encapsulates this text concerning the image or page G , which have its own geometry $\vec{p} \in \mathbb{N}^{W(n), H(n)}$, where $W(n)$ and $H(n)$ are the width and height of each page n , $1 \leq n \leq N$, respectively. It should be noted, however, that the geometry of the regions is optional but an intermediary step towards ultimately obtaining its text.

Additionally, we assume that all acts D^k are contiguous and ordered, and hence the regions R_q are as well. In Figure 4.1, we illustrate the applied nomenclature with examples from the first two and the last images of the Nesle corpus. The figure demonstrates that the first deed, denoted as D^1 , spans two images – G_1 and G_2 – and encompasses four regions, specifically R^1, R^2, R^3, R^4 . The subsequent deed, D^2 , commences in the fifth region, R^5 , and concludes in one of the adjacent pages. The final image in the figure shows the conclusion of deed D^K , which occupies multiple images and several regions.

4. Act Segmentation and Layout Analysis

The problem can be solved by finding the $K + 1$ boundaries β_1, \dots, β_K , $\beta_k \in \mathbb{N}$, $\beta_k \leq \beta_{k+1}$, $0 \leq k \leq K$, with $\beta_0 = 0$ and $\beta_K = Q$. Then, all regions R_Q must be found, given that each boundary is located at the end of each region. Therefore, it can be approximated in two different ways:

1. First, find the geometry that encapsulates the region's text, and subsequently transcribe this text following its corresponding "reading order". Once the regions are identified, it is necessary to establish a relationship between contiguous regions to determine which of these detected regions is described as a boundary of an act. In other words, indicate if a region suggests that an act ends and, consequently, the next act begins in the following region.
2. Alternatively, transcribe or directly obtain all the textual information from the pages and establish a series of markers or "tags" to indicate these boundaries. Similar to the challenges faced in Named Entity Recognition (NER), these "tags" are added in the transcription. A relationship must also be established among these tags to define a boundary or "cut-off" region between text and another.

In both approaches, we delineate the deeds using their respective boundaries. This can be achieved by identifying the geometric contours that envelop the text or assigning specific tags directly to the text. Subsequently, the objective is to model the likelihood of a sequence of regions or labels that characterize these regions, denoted as $R_q \in R_Q$. Regions can be categorized into three types: "Initial" regions (I), which mark the beginning of a deed; "Final" regions (F), which signify the end of a deed; and "Mid" regions (M), which are situated between I and F and serve to continue the deed. It is important to note that, by definition, these "Mid" regions (M) always span an entire column or page. Lastly, we address the scenario where deeds both commence and conclude on the same page or column. We refer to these as "Complete" deeds, denoted by C. We define these labels as $\mathcal{C} \stackrel{\text{def}}{=} \{I, M, F, C\}$. According to these rules, a sequence $c_1, \dots, c_Q, c_j \in \mathcal{C}, 1 \leq j \leq Q$ of region labels must be *consistent*.

To derive the deed segmentation from this label sequence c_1, \dots, c_Q , we need to establish boundary markers b_k at regions labeled as F or C. This is further elaborated in the following pseudocode:

$$\beta_0 := k := 0; \text{ for } (j := 1 \dots Q) \text{ if } (c_j = \mathbf{F} \parallel c_j = \mathbf{C}) \{k := k + 1; \beta_k := j\}; K := k \quad (4.1)$$

4.1.1 Proposed Approaches

The objective is to derive a sequence with maximum likelihood by modeling the likelihoods associated with the classified regions. To accomplish this, we propose three different approaches. The probability of a sequence of labels c_1, \dots, c_N for a book $B = G_1, \dots, G_N$ can be decomposed in the following manner:

$$P(c_1, \dots, c_N | B) = P(c_1 | B) \prod_{j=2}^N P(c_j | B, c_1, \dots, c_{j-1}) \quad (4.2)$$

Under the assumption of Naive Bayes region class independence, and given that $P(c_j | B)$ is solely dependent on the image G_j , we make the following approximation:

$$P(c_1, \dots, c_N | B) \approx \prod_{j=1}^N P(c_j | R_j) \quad (4.3)$$

Subsequently, if we opt for an approach other than Naive Bayes and decompose Eq. (4.2) to include some context, we get:

$$\begin{aligned} P(c_1, \dots, c_N | B) &\approx P(c_1 | R_1) \prod_{j=2}^N P(c_j | R_j, c_{j-1}) \\ &= P(c_1 | R_1) \prod_{j=2}^N \frac{P(c_j, c_{j-1}, R_j)}{P(c_{j-1}, R_j)} \\ &= P(c_1 | R_1) \prod_{j=2}^N \frac{P(c_{j-1})P(c_j | c_{j-1})P(R_j | c_j, c_{j-1})}{P(c_{j-1})P(R_j | c_{j-1})} \\ &\approx P(c_1 | R_1) \prod_{j=2}^N P(c_j | c_{j-1}) \frac{P(R_j | c_j)}{P(R_j)} \\ &= P(c_1 | R_1) \prod_{j=2}^N P(c_j | c_{j-1}) \frac{P(c_j | R_j)}{P(c_j)} \end{aligned} \quad (4.4)$$

4. Act Segmentation and Layout Analysis

In the initial step, two independence assumptions are made: $P(c_j | G_1, \dots, G_N)$ is solely dependent on R_j , and the region's dependency follows a *first-order Markov* model. It is assumed that R_j is conditionally dependent only on c_j and is unconditionally independent of c_{j-1} , leading to $P(R_j | c_{j-1}) = P(R_j)$. In the final step, Bayes' rule is reapplied to express the conditional likelihood $P(R_j | c_j)$ in terms of the posterior probabilities $P(c_j | R_j)$. These decompositions align with a Hidden Markov Model (HMM).

The following proposals aim to segment B into deeds by obtaining a sequence of classes per region with the highest probability:

$$\hat{c}_1, \dots, \hat{c}_N = \arg \max_{c_1, \dots, c_N} P(c_1, \dots, c_N | B) \quad (4.5)$$

where the probability $P(c_1, \dots, c_N | B)$ is approximated using either Eq. (4.3) or (4.4). This process is referred to as *decoding*.

Region Class Modelling

A classifier is required to estimate the class posterior probabilities $P(c | R), c \in \mathcal{C}$, as specified in Eqs. (4.3,4.4). This classifier operates in an entirely local manner, disregarding the context of R —that is, the adjacent pages—and focusing solely on the features of the individual region. The nature of this classifier can vary based on the choices made in the previous section; it could be optical if it uses the images of the detected regions or relies on the text itself.

For a given book B with regions R_1, \dots, R_Q , the classifier is employed to estimate the sequence of class posterior probabilities $P(c | R_j)$. We will refer to this sequence as the book's *posteriorgram*¹ of B :

$$\vec{r}_1, \dots, \vec{r}_N, \quad r_{jc} \stackrel{\text{def}}{=} P(c | R_j), \quad c \in \mathcal{C}, \quad 1 \leq j \leq Q \quad (4.6)$$

Consistency Constraints Model

The probability decomposition of Eq. (4.4) is that of a first-order HMM with a set of states $\mathcal{Q} = \mathcal{C} = \{I, M, F, C\}$ and state transition probabilities

¹Following time-honored tradition in signal processing and automatic speech recognition, the term *posteriorgram* is used for this type of (variable-length) sequences of posterior probability vectors.

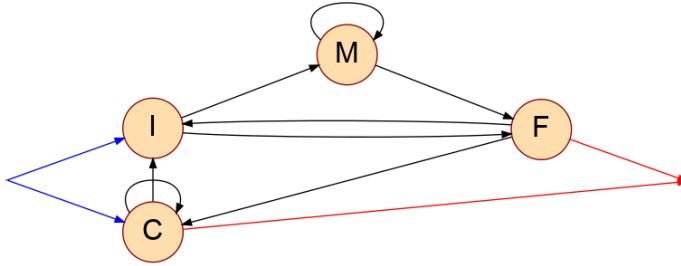


Figure 4.2: Topology of the Consistency Constraints HMM. For convenience, states are labeled with I, C, M, and F, respectively corresponding to the initial, complete, middle, and final regions of a deed.

$P(c | c')$, $c, c' \in \mathcal{Q}$. The state-emission probabilities would be the class-conditional likelihoods $P(R | c)$, where R is a region of an act and $c \in \mathcal{C}$. These likelihoods are proportional to $P(c | R)/P(c)$, as used in Eq.,(4.4), where the posteriors $P(c | R)$ are provided by the classifier (Eq.,(4.6)) and $P(c)$ can be trivially calculated from the GT of the segmented books.

Note that the ultimate goal of segmentation is to preserve the coherence of the textual information of each deed of a bundle. To this end, each deed segment D^k , $1 \leq k \leq K$, must fulfil the following *hard Consistency Constraints* (CC): $R_{b_{k-1}+1}$ must be an I-region or a C-region, R_{b_k} must be an F-region or a C-region and, if $M(K) > 2$, $R_{b_{k-1}+2}, \dots, R_{b_k-1}$ are all M-region.

Correspondingly, only the states F or C can be final and the initial-state probability must be $P(c = I)$ for the state I and $P(c = C)$ for C and 0 for other states. In addition, $P(I | M) = P(M | F) = P(I | I) = P(F | F) = P(M | C) = P(F | C) = 0$. The other transition probabilities can be straightforwardly estimated from GT segmented bundles. This HMM topology is depicted in Figure 4.2.

To apply these restrictions, we use a decoder. Next, we explain different options for decodings.

4. Act Segmentation and Layout Analysis

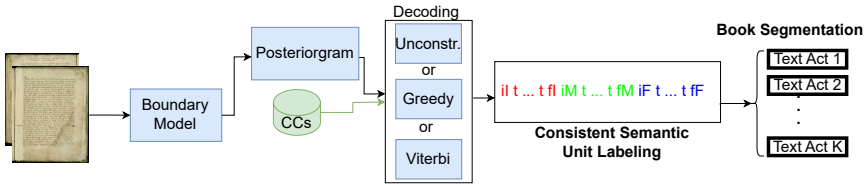


Figure 4.3: Illustration of the proposed comprehensive pipeline. The input consists of images or regions, for which we apply the boundary model. Then, using the CCs, we obtain a consistent sequence using a decoder. Finally, the acts are segmented across the book.

Unconstrained Decoding

If we do not assume the consistency, the optimization of Eq.(4.5) using Eq. (4.4) becomes trivial, obtaining a probably not-guaranteed result:

$$\hat{c}_j = \arg \max_{c \in \mathcal{C}} r_{jc}, \quad 1 \leq j \leq Q \quad (4.7)$$

Greedy Decoding

A consistent solution to Eq.(4.4), although not globally optimal, is a *greedy decoder* locally applying the CCs as follows:

$$\hat{c}_j = \arg \max_{c \in \rho(\hat{c}_{j-1})} r_{jc}, \quad 1 \leq j \leq Q-1; \quad \hat{c}_{Q-1} = \pi(\hat{c}_{Q-2}); \quad \hat{c}_Q = F \quad (4.8)$$

where the function $\rho : \mathcal{C} \rightarrow 2^{\mathcal{C}}$ is defined as: $\rho(I) = \rho(M) = \{M, F\}$; $\rho(F) = \{I, C\}$; $\rho(C) = \{I, C\}$, and $\pi : \mathcal{C} \rightarrow \mathcal{C}$ is a “previous label function” defined as: $\pi(I) = \pi(M) = M$; $\pi(F) = I$.

Although the proposed solution is consistent, it does not guarantee that the probability of the sequence is maximum.

Viterbi Decoding

If we follow Eqs. (4.4) and (4.6) the optimization from Eq. (4.5) becomes:

$$\hat{c}_1, \dots, \hat{c}_N = \arg \max_{c_1, \dots, c_N} \prod_{j=1}^N r_{jc_j} P(c_j | c_{j-1}) \quad (4.9)$$

To achieve a globally optimal solution for this equation, a Dynamic Programming-based decoder is essential. The Viterbi algorithm, a widely used decoder in this context, can be described as follows.

Consider $V(j, q)$ to represent the probability of the max-probability state sequence ending at state q and generating the initial j labels and set $V(1, l) = r_{1,l}$, $V(1, C) = r_{1,C}$, $V(1, M) = V(1, F) = 0$. The subsequent recurrence relation is valid for $1 \leq j \leq N$, and taking into account the “dummy” region define before, equivalent to a F region, $c_0 = F$:

$$V(j, q) = \max_{q' \in \mathcal{Q}} r_{jq} P(q | q') V(j-1, q'), \quad q \in \mathcal{Q} \quad (4.10)$$

where g_{jq} , $q \in \mathcal{Q} \equiv \mathcal{C}$, $1 \leq j \leq Q$ are the components of the posteriorgram of the bundle. Upon computing $V(N, F)$, backtracing provides a globally optimal sequence of states and the associated sequence of I,M,F,C, labels.

Figure 4.3 provides an overview of the entire pipeline, where the “Boundary Model” and the “Decoding” box should be selected from one of the explained options.

4.2 A Whole-Book Evaluation Measure

Upon segmenting the \hat{K} acts of each bundle and obtaining \hat{B} , we seek to determine how accurately they have been segmented compared to the test set. Traditional HTR metrics such as CER and WER could be employed, but these do not indicate the segmentation quality. We are interested in understanding the amount of information lost due to, for example, bisecting an act or concatenating several of them.

Furthermore, there is a distinction between segmenting an act with an extra page containing little to no text and an act with an additional page densely filled with information. Although both instances are errors, the former is considerably less severe, while the latter is a severe error and should be penalized accordingly.

4. Act Segmentation and Layout Analysis

To tackle these issues, we should contemplate developing a metric that measures segmentation quality by accounting for the severity of errors, such as splitting or merging acts, regarding information loss caused by poor segmentation. This metric should penalize severe mistakes, such as including a part of a page brimming with information into an incorrect act, more heavily than less serious ones, such as misaligning a chunk without text from another page. By creating and implementing a metric of this nature, we can more effectively assess the efficacy of our method in segmenting books and identify areas for enhancement.

We propose a metric called the Content Alignment and Error Rate (CAER) to quantify the amount of information lost due to improper segmentation and inaccurate text recognition. Therefore, we must evaluate both the obtained textual information and the segmentation and alignment between the hypothesis and reference sets.

The initial step entails generating a Bag of Words vector (or *running words* vector) $\vec{D} \in \mathbb{R}^N$ for each act's text in \hat{B} . Here, N signifies the total count of words in both the reference and hypothesis sets. The Figure 4.4 represents that pipeline that transforms an act D^k (a series of ordered regions already segmented) to a BoW vector using PrIx. This is done for every segmented act to be able to calculate the CAER.

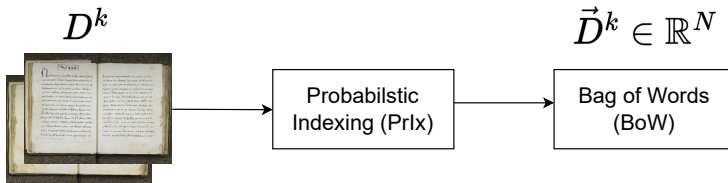


Figure 4.4: Illustration of the pipeline from an act D^k to BoW using PrIx.

From now on, we have the hypothesis of the book segmentation made by the system represented in a sequence $\vec{D} = \vec{D}^1, \dots, \vec{D}^K$, and its corresponding GT reference as $\vec{D} = \vec{D}^1, \dots, \vec{D}^K$. In other words, each segmented act corresponds to a feature vector based on its textual content. Now we have to see how we can evaluate it against the reference vectors in GT. We can address this problem by calculating the minimum number of operations to transform D into \hat{D} using dynamic programming, following the subsequent recurrent relation:

$$\begin{aligned}
 E(i, j) = \min & (E(i, j - 1) + L(\vec{0}, \vec{D}^j), \\
 & E(i - 1, j - 1) + L(\vec{D}^i, \vec{D}^j), \\
 & E(i - 1, j) + L(\vec{D}^i, \vec{0}))
 \end{aligned} \tag{4.11}$$

where $\vec{0}$ is an “empty vector” and $L(X, Y)$ is the “Bag of Words Error Rate” (bWER) distance between X and Y , defined in [TV23].

As a result of solving Eq. (4.11), we obtain the edit operations as a result of the three min terms of the function, interpreted as *deed edit operations*: insertions, substitutions, and deletions, respectively corresponding to the three terms of the min function.

An insertion indicates when an act appearing in the hypothesis does not appear in the reference. That is, it has been erroneously inserted into the hypothesis. The cost assigned to such an insertion depends on the textual content and is calculated as $L(\vec{0}, \vec{D}^j)$, which uses the total number of running words that are not in D_j and must be inserted. The deletion occurs in a similar way but in reverse, where an act \hat{D}^i has been poorly recognized and has a cost $L(\vec{D}^i, \vec{0})$, which is the total number of running words in \hat{D}^i . For the case of substitution, the bWER distance between the two aligned acts is calculated as $L(\vec{D}^i, \vec{D}^j)$.

If $D, \hat{D} \in \mathcal{D}$ represent a pair of reference and hypothesis deeds, then $L(D, \hat{D})$ is conceptually akin to the bWER as described in [Vid+23], albeit without normalization. This metric estimates the number of *word* insertion, substitution, and deletion operations required to transform the text in D into that of \hat{D} . Importantly, it does so while *disregarding* the *order of words* in either D or \hat{D} . Specifically:

$$L(D, \hat{D}) \stackrel{\text{def}}{=} \frac{1}{2} \left(\|\vec{D} - \vec{\hat{D}}\|_1 + \left| \|\vec{D}\|_1 - \|\vec{\hat{D}}\|_1 \right| \right) \tag{4.12}$$

Finally, for the reference deed sequence of a bundle $B = D^1, \dots, D^K$ and the corresponding segmentation hypothesis $\hat{B} = \hat{D}^1, \dots, \hat{D}^{\hat{K}}$, the CAER is defined as:

$$\text{CAER}(D, \hat{D}) = \frac{1}{\mathcal{W}} E(\hat{K}, K) \tag{4.13}$$

where $\mathcal{W} = \sum_{j=0}^K \|\vec{D}^j\|_1$ is the total number of the N -selected running words for the bundle.

4. Act Segmentation and Layout Analysis

With CAER, we have achieved the previously discussed objective, measuring the extent of textual information loss when segmentation fails. In this manner, if a mistake is made when segmenting a chunk of a page with insignificant content, the impact on the metric is minimal. However, if an error occurs while segmenting a page with much textual information, it is notably reflected in the metric, contingent upon the size of the act. Similarly, the metric considers the size of the acts, not merely the number of accurately segmented acts.

It is important to notice that with this metric, as it calculates an error, the closer the results are to 0, the better the performance. It is also worth noting that the Viterbi algorithm is expected to outperform a Greedy algorithm under usual circumstances. However, in this particular situation, Viterbi aims to maximize the sequence probability while adhering to the CCs, but it does not optimize the metric we have just introduced, the CAER. Nevertheless, a high correlation is anticipated concerning segmentation, although this is not assured.

4.3 Restricted Multi-page Act Segmentation

In this section, we segment the AHPC books (see Appendix A.2.1). Due to the restrictions of this corpus, we assume that acts are physically separated at the page level. This means that if an act were to end halfway down a page, for example, the other half of the page cannot contain another act but would be left blank, and the next act would start on the following page.

In addition, we make another assumption, and following the rules used when the AHPC corpus was written (see Appendix A.2.1), every act occupy at least two consecutive pages. So from now on, due to the characteristics of the corpus, a region be the same as a page.

Taking these assumptions into account, from now on, let be the set of page-classes $\mathcal{C} \stackrel{\text{def}}{=} \{I, M, F\}$, similarly to the previous section and refer to the starting and ending pages as “Initial” (I) and “Final” (F), respectively. Similarly, we refer to all pages between I and F as “Mid” (M). There is no C class in this corpus since all acts have, at least, a length of two pages.

Additionally, as bundles may contain images without content or simply blank, we refer to these as “junk”. These can appear anywhere within the

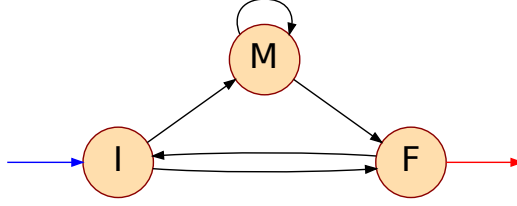


Figure 4.5: Markov chain of the Consistency Constraints, where I, M, and F correspond to the beginning, middle, and end of an act, respectively.

bundle, with no restrictions, whether between acts or at the beginning or end of one. However, junk pages are almost trivial to detect. Therefore, for simplicity, we assume that we have been able to remove them from the bundle in a previous step. To do this, we have calculated the running word for every page within the bundle separately and removed those with a running word lower than σ . We have summed up all the running words to calculate the running word for a page **D**. Following Section 2.1.2.1 and using the PrIx of the collection (see Section 3.2.1), it has been calculated as follows:

$$f(\mathbf{D}) = \sum_{v \in V} f(v, \mathbf{D}) \quad (4.14)$$

where **D** is the document and V is the same vocabulary used in [Pri+21; Flo+22]. In that case, we use all of the running words without any distinction.

As explained in Section 4.1, let $B = D^1, \dots, D^K$ be a bundle which sequentially encompasses K acts or deeds. Each act D^k , in this case, is a sequence of $M(k) \geq 2$ regions, as defined in Section 4.1, denoted as $D^k = G_{k_1}, \dots, G_{k_{M(k)}} \equiv R_{k_1}^k, \dots, R_{k_{M(k)}}^k$. In this case, it is the same as non-junk regions or page images. Now, following the problem definition from Section 4.1, the problem is to find the $K+1$ boundaries b_k , $0 \leq k \leq K$, but with the following restrictions: $b_0=0$, $b_{k-1} < b_k$, $b_K = M \equiv Q$, and the bundle becomes described as a sequence of deeds $D = D^1, \dots, D^K$, where $D^k = G_{b_{k-1}+1}, \dots, G_{b_k} \equiv R_{b_{k-1}+1}, \dots, R_{b_k}$ and $M(k) = b_k - b_{k-1} + 1$, $1 \leq k \leq K$.

Additionally, each page G_b , $1 \leq b \leq N$, must satisfy the following CCs: $G_{b_{k-1}+1}$ is an I-page, G_{b_k} is an F-page, and if $M(K) > 2$, $G_{b_{k-1}+2}, \dots, G_{b_k-1}$ are all M-pages. This has been represented as a Markov

4. Act Segmentation and Layout Analysis

Chain in Figure 4.5, which is slightly different to Figure 4.2, where G-state has been removed.

Then, a sequence c_1, \dots, c_N , $c_j \in C$, $1 \leq j \leq N$ of page labels that follows these rules is said to be *consistent*. Given the label sequence c_1, \dots, c_N , the corresponding segmentation is readily obtained by successively setting the boundaries b_k to the positions of the pages labeled with F, as outlined by the following pseudocode:

$$b_0 := k := 0; \text{ for } (j := 1 \dots N) \text{ if } (c_j = F) \{k := k + 1; b_k := j\}; K := k \quad (4.15)$$

To address this issue, we propose using a page (region) classifier in the first instance, then *decoding* the whole sequence utilizing the posterior probabilities to achieve an output consistent with the CCs. That is, following Eqs. (4.3) or (4.4).

4.3.1 Visual Image Classifier

We have experimented with various neural network-based models for such classifiers, including convolutional neural networks such as ResNet- $\{18, 50, 101\}$ [He+15], ConvNeXt [Liu+22], and transformer-based models like Swin [Liu+21]. Rather than randomly initializing these models' weights, we used pre-trained models on ImageNet [Wol+20b].

ResNet and ConvNeXt are convolutional neural network-based models (see Section 2.2.2) with a final linear layer employed to obtain a posterior probability for each class $c \in \{I, M, F\}$. ResNet is a widely used architecture for image classification tasks and more. Its residual connections² enable the training of deeper neural networks. The architecture of such networks is often determined by the number of blocks connected with these connections. A higher number of blocks leads to a larger and deeper network and more parameters to train.

Additionally, we have ConvNeXt, which, as the authors describe [Liu+22], is a “modernized” version of ResNet with the latest advancements in ConvNet training, also driven by significant progress in Vision Transformers in

²A residual connection in a neural network is a direct link that allows information to bypass one or multiple layers in the network, facilitating gradient flow during training and mitigating the vanishing gradient problem.

recent times. In our case, we utilized ConvNeXt base, an intermediate-sized model. We tested smaller models, such as ConvNeXt tiny, but the results were inferior.³ We also attempted to train Vision Transformers, such as Swin [Liu+21]. However, these models faced many convergence issues, and the results were considerably worse than the others. We believe a possible explanation for this is, besides the general sensitivity of transformers to hyperparameters, that the pre-training of these models (on ImageNet) is performed with 224×224 resolution images. In contrast, in this work, we need to use resolutions of at least 1024×1024 . This is because, as seen in previous sections, we need to retain fine-grained details in the images to classify them correctly, such as boxes or words.

Note that this classifier ignores the image’s context, not considering the surrounding pages. Ultimately, once we have the trained classifier, we obtain the class-posterior estimate $P(c \mid R)$, $c \in \mathcal{C}$ for each region (or page-image) R in a test book $B = R_1, \dots, R_Q$, following Eq.(4.5).

4.3.2 Obtaining a consistent segmentation

As explained in Section 4.1, the terms $P(c_j \mid G_j, c_{j-1})$ of Eq.(4.4) can be interpreted as the transcription probabilities of a first-order Markov chain with a set of states $\mathcal{Q} = \mathcal{C} = \{I, M, F\}$.

Since we want to be able to retrieve the information from each deed of the bundles, let us remember that segmentation has to preserve the coherence of the textual information. To do this, each deed segment, D^k , $1 \leq k \leq K$, must comply with the following CCs, which have a slight modification with respect to the CCs presented in Section 4.1: $R_{b_{k-1}+1}$ is an I-region, R_{b_k} is an F-region, and if $M(K) > 2$, $R_{b_{k-1}+2}, \dots, R_{b_k-1}$ are all M-regions. Note that a region is equivalent to a page due to the corpus restrictions.

Therefore, the sequence must start with a state I, so the initial probability for I must be 1 and 0 for the other states. Similarly, the last state should be F. Also, $P(I \mid M) = P(M \mid F) = P(I \mid I) = P(F \mid F) = 0$. The other transitions can be easily estimated from the reference GT. This pipeline is shown in Figure 4.6, where the *Boundary Model* is now a *Page Level Classification* and the CCs have one less state.

³It is worth noting that larger models are quite resource-intensive, so we could not test others.

4. Act Segmentation and Layout Analysis

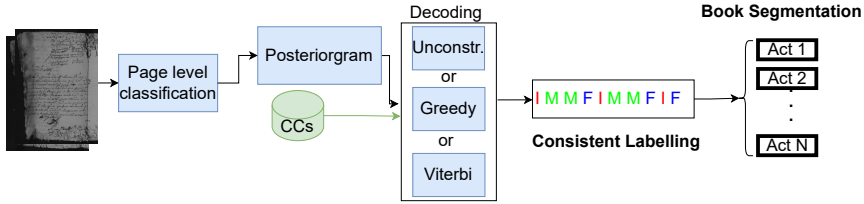


Figure 4.6: Illustration of the proposed comprehensive pipeline. The input consists of images, from which we classify and obtain the posterior probabilities for each class $c \in \{I, M, F\}$. Subsequently, by employing the CCs, we derive a consistent sequence using one of the two alternative decoding approaches.

4.3.3 Evaluation

Once we have segmented all the \hat{K} acts of each bundle and obtained \hat{B} , we would like to evaluate it using the test set. Without applying any of the consistency rules explained in the previous section, or any decoding, we can evaluate the classification error on the different page-images separately. Also, next to this we would like to know how well is segmenting the acts in an end-to-end manner.

4.3.3.1 Assesing Visual Image Classifier

Often, classifiers like those shown in Section 4.3.1 are evaluated using the conventional classification error, where the class with the highest probability is chosen over the hypothesis. However, our approach diverges from this norm. Instead of solely relying on the class with the maximum probability, we employ a decoder that considers the probabilities of each page collectively. As a result, traditional error metrics become less relevant for our evaluation.

In this case, we are interested in knowing how well-calibrated the probabilities are for each class. To do this, we compare the output posterior distribution with the reference distribution using *cross-entropy*:

$$H(P_t, P) = -\frac{1}{Q} \sum_{j=1}^Q \sum_{c \in \mathcal{C}} P_t(c | R_j) \log_2 P(c | R_j) = -\frac{1}{Q} \sum_{j=1}^Q \log_2 r_{j,c_j} \quad (4.16)$$

where P_t is the “target distribution”, defined as $P_t(c | R_j) = 1$ iff R_j is of class c , according to GT, $Q \equiv N$ is the total number of samples (pages or regions) and r_{j,c_j} is the class-posterior of the image $G_j \equiv R_j$ for the reference class c_j , as defined in Section 4.3.1.

4.3.3.2 Assessing Bundle Segmentation Performance End-to-End

However, with cross-entropy, we are not evaluating the final purpose we want, which is to know how well a book is segmented, although it can help us decide which model to stick with to carry out the complete segmentation process.

Bundle Segmentation Error Rate

In our current problem, a deed is conceptualized as a collection of pages. This allows us to devise a straightforward method for comparing two deeds without considering their textual content.

Defining the sets of reference and hypothesis images as $D^i, \hat{D}^j \in \mathcal{D}$, respectively, we can calculate the cost of individual alignment as the *symmetric set difference* $L(D^i, \hat{D}^j) \stackrel{\text{def}}{=} |D^i \ominus \hat{D}^j|$, which can be calculated as follows:

$$L(D^i, \hat{D}^j) = |D^i \cup \hat{D}^j| - |D^i \cap \hat{D}^j| \quad (4.17)$$

Therefore, following this definition, we can calculate the cost of insertion as $L(\epsilon, \hat{D}^j) = |\hat{D}^j|$, which is the number of pages in \hat{D}^j . Deletions can be calculated similarly as $L(D^i, \epsilon)$, where it is the number of pages in D^i . And the cost of a substitution $L(D^i, \hat{D}^j)$ is just the number of page images which are in D^i but not in \hat{D}^j plus are in \hat{D}^j but not in D^i .

Finally, for the reference deed sequence of a bundle $B = D^1, \dots, D^K$ and the corresponding segmentation hypothesis $\hat{B} = \hat{D}^1, \dots, \hat{D}^{\hat{K}}$, the Bundle Segmentation Error Rate (BSER) is defined as:

$$\text{BSER}(D, \hat{D}) = \frac{1}{T} E(K, \hat{K}) \quad (4.18)$$

where $E(K, \hat{K})$ is computed using Eq. (4.11) with the cost function given by Eq. (4.17) and $T \stackrel{\text{def}}{=} \sum_{i=0}^K |D^i|$ is the total number of page images in B .

Content Alignment Error Rate using PrIx

We also should consider designing a metric that evaluates the quality of the segmentation while taking into account the severity of errors, such as splitting acts or merging them. We use the metric explained in Section 4.2 to achieve this evaluation.

However, as we lack transcribed text and ground truth text, we employ the PrIx method [Pui18] to generate the $\vec{D} \in \mathbb{R}^n$ vectors. We also apply the IG approach to select the pseudowords for representation, as detailed in Section 2.1.2.2, always choosing the top $n = 16384$ with the highest IG in this instance. The n components of these vectors correspond to the n words with higher IG, as determined in [Pri+21; Pri+23b], and the value of each component is the expected number of occurrences of the corresponding word, estimated from the image PrIx as also discussed in [Pri+21; Pri+23b]. In the scope of these studies, a total of 12 distinct classes of deeds were taken into account for the calculation of IG within the same corpus.

For each component, we have calculated the number of occurrences of that component v in the document D , $f(v, D)$. This representation allows us to compact the most relevant textual information from a set of images that form an act. Finally, following Eqs.(4.11) and (4.12), the CAER can be computed.

4.3.4 Empirical Settings and Results

In this subsection, we discuss the results obtained for both classification and full book segmentation. However, before delving into the results, the empirical settings used to replicate these are explained.

4.3.4.1 Empirical settings

For these segmentation experiments, we use the AHPC corpus. Given that in a real-world (production) scenario, the most costly aspect is data labeling, we aim to minimize the training data while increasing the test data. Therefore, we obtain results by training with just one book and testing the remaining three, performing all four possible combinations with the four books we have from AHPC. Additionally, in a subsequent experiment, we limit the number of deeds per book we have available for training, thus obtaining a learning curve with respect to the number of training samples.

Be aware that the metrics analyzed in Section 4.3.3.2 are established at the bundle level. For instance, in Eq.(4.13), CAER represents the cost of all editing operations for deeds within a single bundle, normalized by the bundle’s total page count. In the framework of our proposed protocol, each experiment entails testing with three distinct bundles. To calculate the metrics, we employ a micro-averaging approach, where the cost is accumulated over the three bundles and finally normalized by the total number of pages of these bundles.

We have set $\sigma = 20$ to apply the first preprocessing step and remove junk pages. Subsequently, all the image classifiers explained earlier have been trained for at least 15 epochs and a maximum of 30, with an early stopping of 5 epochs based on an evaluation set. This evaluation set has been the same for all models and has been randomly selected using 15% of the training set. A learning rate of 0.001 has been employed with AdamW as the optimizer [LH17], with a decay of 0.5 after every 10 epochs. The batch size is set to 4 except for the larger model, ConvNeXt, which is reduced to 2 due to memory limitations. All images have been resized to 1024×1024 .

To mitigate fluctuations due to the initialization of learning algorithm parameters, all values reported in tables and curves in Section 4.3.4.2 are averages of results obtained with 10 random parameter initializations.

4.3.4.2 Results

In Table 4.1, we evaluate the image classifiers separately, as explained in Section 4.3.3.1, measuring the quality of each one’s class-posterior distribution. That is, using cross-entropy as the measure for each book, as well as an average of the four results.

Table 4.1: Cross-entropy (bits/page) between the hypothesis page-image class posterior and the reference (0/1) probability distributions. Training with one bundle and testing with the other three bundles.

Classifier	JMBD4946	JMBD4949	JMBD4950	JMBD4952	Average
ResNet18	0.028	0.027	0.031	0.116	0.050
ResNet50	0.013	0.009	0.014	0.022	0.015
ResNet101	0.043	0.016	0.028	0.065	0.038
ConvNeXt	0.017	0.009	0.022	0.027	0.019

4. Act Segmentation and Layout Analysis

We see that the best result (the one with the lowest cross-entropy) is offered by ResNet50, followed by ConvNeXt. Following these results, the next steps are taken only with the ResNet50 model. It is important to note that no decoder has been applied at this stage, making the results preliminary and unsuitable for segmentation tasks based on maximum probability classes.

Once the best model for classifying the pages (ResNet50) has been chosen, we can obtain, using the posteriorgrams produced by this model, and using one of the decoders proposed in Eqs.(4.7), Eq. (4.8), and Eq. (4.9) from Section 4.3.2. From this segmentation, we obtain the BSER and CAER results, shown in Table 4.2.

We can observe that the trend is for BSER to be slightly higher than CAER. This is usually the case because BSER gives the same importance to all pages, while CAER does not. As explained in Section 4.3.3.2, CAER penalizes more a page that is poorly segmented and has more textual content than a blank page. In fact, a completely blank page would not be penalized due to not having textual content.

As anticipated, the Viterbi decoder outperforms other decoders across all four experiments. In the last column, which presents the average across the four books, Viterbi's performance is up to three times superior to other decoders.

The most favorable results are achieved when training with the book JMBD4946, yielding a BSER of 5.6% and a CAER of 4.5%. Conversely, the highest error rates are observed when training with JMBD4952, resulting in a BSER of 13.4% and a CAER of 10.8%. It might be interesting to analyze where some of these errors occurred. In Figure 4.7, we can see the two images typically were misclassified after the Viterbi decoding process. They caused the incorrect segmentation of two acts, resulting in two insertions and two substitutions. Both images should have been classified as class $\hat{c} = M$.

These results align with the average number of pages per deed, showing that training with larger deeds leads to a lower segmentation error on the test bundles. Each of the first three bundles has an average of close to 6 pages per deed, leading the low segmentation errors. In comparison, JMBD4952, with 4.2 pages per deed (the lowest count), exhibits the highest segmentation error.

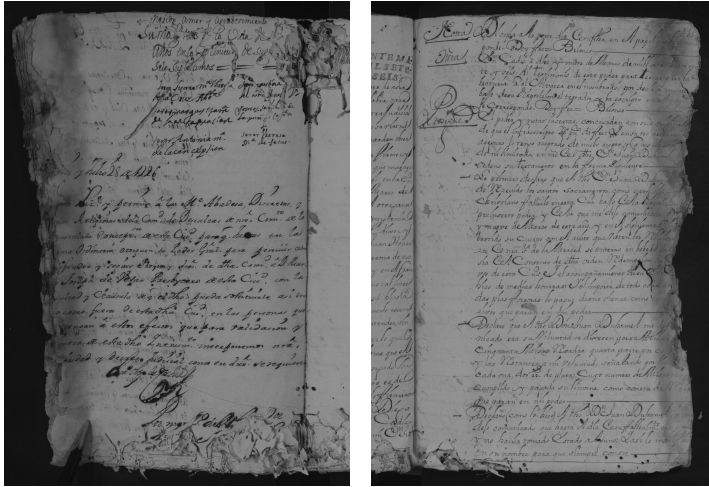


Figure 4.7: Pages 717 and 888 of the JMBD4952 book, respectively. The first page belongs to the $\hat{c} = F$ class, while the second one belongs to $\hat{c} = M$ class. The first has been misclassified as class $c = M$, while the second has been misclassified as class $c = F$.

Table 4.2: BSER and CAER achieved by different decoders, training with one bundle and testing with the other three bundles. The page image classifier was ResNet50. Results are in percentage.

Metric	Decoder	JMBD4946	JMBD4949	JMBD4950	JMBD4952	Average
BSER	Unconstrained	23.3	24.5	23.3	37.1	27.0
	Greedy	22.7	24.5	23.1	36.2	26.2
	Viterbi	5.6	9.0	5.7	13.4	8.4
CAER	Unconstrained	19.1	20.1	21.0	29.4	22.4
	Greedy	18.4	20.1	20.6	28.6	22.0
	Viterbi	4.5	7.4	4.8	10.8	6.9

Finally, to test the reliability and robustness of the system, we trained the best model presented in Table 4.1, ResNet50 using the Viterbi decoder, using an increasing number of deeds for each book. Instead of training the model with an entire book, we only used a set number of deeds from that book and tested with the other three books, as we have done previously. These deeds were chosen in powers of two, respecting the order in which they appear in the book, without repetition. Furthermore, to avoid random initialization effects and to obtain more consolidated results, each test was

4. Act Segmentation and Layout Analysis

repeated 10 times, and an average was taken.

The results can be seen in Figure 4.8. We observe that the more training samples we have, the better results we achieve. Notably, from 64 deeds onwards, we begin to obtain results with less than 20% segmentation error, except for JMBD4952. With 128 deeds, we achieve, if not the best, results very close to the best ones obtained. The trend at the end of the curve for the book JMBD4952 is particularly striking, suggesting that we might achieve even better results with more data.

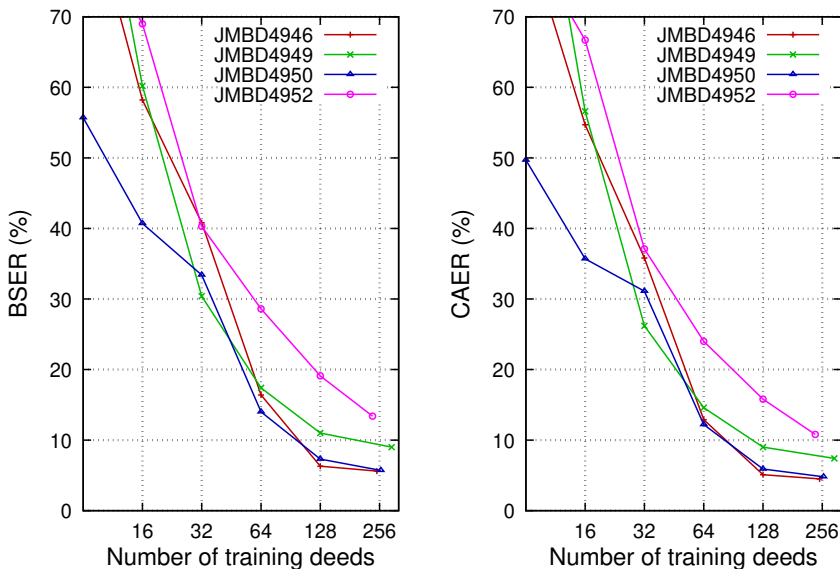


Figure 4.8: Learning curve when training with each bundle. Page class posteriors (posteriorgrams) were obtained with a ResNet50 classifier and decoding was carried out with the Viterbi algorithm.

In conclusion, combining deep learning classifiers and decoding techniques, such as the Viterbi algorithm, proves promising for book segmentation. The ResNet50 mainly shows notable performance in book classification and segmentation. While errors do exist, they are minimal and locatable, suggesting that there is room for improving and refining this method but probably using other corpora with more challenging data.

Therefore, this work represents a significant step towards automating the book segmentation process at the act level with the assumptions considered.

Not only does it promote efficiency, but it also ensures consistency and accuracy, which is essential in extracting information from historical books and documents.

4.4 Multi-page Act Fine-Grained Segmentation

Until now, we have segmented books assuming some rather significant constraints, where each act ended on one page, and the next had to begin on another, and all acts spanned at least two pages. These constraints have simplified the problem, creating an equivalence between regions and pages, and allowing us to test page-level classification methods to solve the issue.

However, although there are large volumes of data with these characteristics, we cannot always assume these act-level restrictions. For this reason, in this subsection, we introduce another method for segmenting acts, called *Fine-Grained Segmentation*, without the hard constraints used in the last section. Furthermore, as we have the GT transcriptions of the corpora, we not only segment but also transcribe the acts simultaneously within the same system.

The system or pipeline presented in the following section, which obtains the segmentation of the entire book into acts and their transcription, consists of two main steps quite similar to what we have done in the previous section. The first step operates at the page level, while the second is responsible for consolidating all the page-level results using almost the same algorithms we have previously employed for decoding. Additionally, we anticipate that we test two approaches to carry out the first step to compare and identify each method's shortcomings.

4.4.1 Processing page by page

The problem we aim to solve is segmenting and transcribing a book into acts. In this subchapter, we also refer to acts as semantic units, which are logical structural units that can span from a paragraph to several pages but share common or related content. As we mentioned earlier, due to the physical constraints of the medium on which the acts are written, sometimes a single paragraph was enough to start and end an act, while other times several paragraphs were needed, using more than one page or column. Consequently, acts constitute semantic units that should be connected and

4. Act Segmentation and Layout Analysis

analyzed as a single item. An example of this, outside of ancient documents, would be newspaper articles, which can span multiple pages. However, if we wanted to search for information about them, we would be interested in treating them as a single block of text rather than separate blocks divided physically. In the case of ancient documents, this mainly occurs in notarial acts. Some examples of notarial acts spanning different pages have been shown in Figure 3.5 and Figure 3.6.

We work at the page level as a first step to address this problem, with two different proposals to obtain the segmentation of acts and their transcription at this level. One involves using an end-to-end system to handle this process in a unified step. A second approach consists of different subsystems: one for detecting and classifying acts on each page and another for obtaining their transcription. However, regardless of the method used, the output of this first step is the segmentation of the page at the act level along with their transcriptions.

In a manner closely analogous to the previous one, we continue to utilize the labels I, M, and F. However, we also add a label, C (Complete), to refer to an act that begins and ends within the same paragraph or text unit on the same page. An instance of the usage of C can be seen in Figure 3.6, where there are as many as eight C's, indicating that there are eight acts for which all of their textual information is contained within the same paragraph. This means we return to having CCs, but they slightly change compared to the previous ones, as shown in Figure 4.2. Moreover, as the previous restrictions no longer apply, we no longer have junk pages to discard.

The following explains the two proposed alternatives for working at the page level. However, although there are two alternatives, the output of both systems is the same. The goal is to transcribe the text and search for a series of “mark-up tokens” representing the “IMFC” states and help delimit and segment the different acts within a book, to finally segment the book in K notarial acts, $D = D^1, \dots, D^K$. These mark-up tokens are defined as $S \subset \Sigma$, $S = \{iI, fI, iM, fM, iF, fF, iC, fC\}$. The characters I, M, F, C denote *initial*, *medium*, *final* and *complete* text block, respectively, while *i* and *f* allow us to distinguish between an opening markup and the corresponding closing. This set of tokens S corresponds to the C labels proposed for usage in Section 4.1 and the purpose of these is to delimit in the text the R_1, \dots, R_Q regions explained in Section 4.1.

In both alternatives, we have a series of N pages ordered by book, $B = G_1, \dots, G_N$, and a set Σ of characters to predict, with $t = |\Sigma|$. However, Σ is slightly different in each alternative, as we will see. The output of both systems is, for an image G_i , the posteriorgram H^i , which is a sequence of posterior probability vectors for characters, $H^i = \vec{H}_1^i, \dots, \vec{H}_{T_i}^i$, where T_i is the number of tokens predicted in image i . Each $\vec{H}_j^i, 1 \leq j \leq T_i$ is a t -dimensional vector indexed by the characters in Σ . Ultimately, the result is a concatenation of all the posteriorgrams at the page level into a single posteriorgram at the book level, $H = H^1, \dots, H^N$.

4.4.1.1 An End-to-End model at page level

One of the solutions we propose for the first step at the page level is to treat it as a purely HTR task. This means using an end-to-end model at the page level, obtaining the transcription and, simultaneously, the mark-up tokens alongside the text. By doing this, we do not need the physical geometry of each region but rather their logical order and position within the page. We refer to the process of transcribing the page and obtaining the mark-up tokens of the layout as *Extended Recognition*.

We use DAN [CCP23] for the HTR model. DAN is a model based on a CNN as an encoder and a transformer decoder, which can leverage both visual and textual content at the page level. Although we have chosen DAN because it can work at the page level without requiring the physical layout, this step could be performed with any other HTR model. For example, the problem could be approached using the classic HTR pipeline, which involves line segmentation and a separate line-level model to obtain transcriptions.

It is worth mentioning that to train DAN, following the original work [CCP23], it is necessary to create synthetic data similar to the target data so that the model can first transcribe at the line level and then move on to the paragraph level. It is essential to understand the curriculum learning process of the model, which is based on transcribing one line of text at first and then moving up to a maximum number of lines of text per corpus. This is done using synthetic data, not the “real” data. For this part, we modified the SynthTiger tool [Yim+21] to create documents that closely resemble the training set. We used fonts similar to the handwritten ones and Latin and

4. Act Segmentation and Layout Analysis

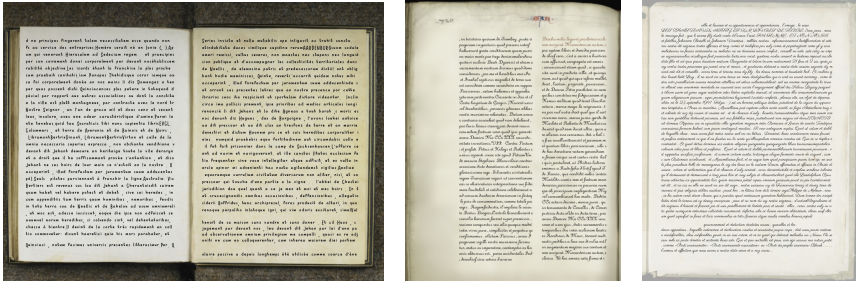


Figure 4.9: Examples of synthetic data following the layout observed in the training set for Nesle, Denis, and Navarre, respectively.

French text from Wikipedia for text generation. We provide an example for each corpus in Figure 4.9.

Once we obtain the posteriorgrams $H^i = \vec{H}_1^i, \dots, \vec{H}_T^i$ for each image G_i , we concatenate them all to create a book-level posteriorgram, $H = H^1, \dots, H^N$.

4.4.1.2 A pipeline at page level

The other alternative we propose for working at the paragraph level is to create a pipeline with the Q regions associated to paragraphs each one. Then, each paragraph has a class $c \in \{I, M, F, C\}$. Subsequently, these paragraphs or cropped pieces of text are transcribed by an HTR model. This, of course, implies having the physical geometry of the acts at the page level, whereas, in the previous proposal, this was not necessary.

Following the definition given in [Qui22], region detection is defined as the search for the most probable layout \hat{h} in image G_i ,

$$\hat{h}_i = \arg \max_{h_i \in H} P(h_i | G_i) \quad (4.19)$$

where H is the set of all possible layouts. Each layout h_i is defined as a set of regions, $h_i = \{R_j, \dots, R_{j+k}\}$, where j is the offset of the region respect to the Q ordered sequences of the book, k is the number of regions in image G_i , and each region, $r_i = \{\vec{p}, c\}$, where $\vec{p} \in \mathbb{N}^4$ is the rectangle definition and $c \in \{I, M, F, C\}$ is its class. The proposal, then, is to find exactly the Q regions and tag them as seen in Figure 3.5 and Figure 3.6. There are numerous object detectors in the literature, but we focus on using

one based on RPN Section 2.2.2.1. Due to its versatility and low resource usage, we have chosen MaskRCNN [Ren+17].

At the same time, an HTR model is trained at the paragraph level. For simplicity, we have used DAN, the same as in the previous subsection, which can also work at this level. It was decided to train this model from scratch and create synthetic data at the paragraph level, very similar to the page-level data, using the same tools for this purpose: the same Latin and French texts and SynthTiger. This model obtains the paragraph-level posteriorgrams very similarly to the previous subsection, with only a slight difference: the mark-up tokens are not transcribed. The posterior probabilities of these mark-up tokens are added depending on the class obtained by the object detection model for each region.

Therefore, once the acts are detected with the object detector and transcribed with the HTR model, we combine both results and obtain the posteriorgrams for each image i , obtaining a sequence of posteriorgrams $H^i = \vec{H}_1^i, \dots, \vec{H}_T^i$. Note that now the set of characters Σ differs a bit from the one used in Section 4.4.1.1, since we do not need the mark-up tokens. These tokens are set by the RPN model at the start and end of each detected region automatically. Finally, following the same steps as in the previous subsection, all the page-level posteriorgrams are concatenated to create a book-level posteriorgram, $H = H^1, \dots, H^N$.

It is noteworthy that by deploying a segmented pipeline like this, we can independently evaluate each model, thereby enabling a more precise determination of strengths and weaknesses within the pipeline. This means we can discern whether a greater degree of information loss is attributable to act detection or transcription, which is considerably more challenging when working within an end-to-end system.

4.4.2 Processing a full book

A book is a sequence of pages but is also considered a sequence of K semantic units, $D = D^1, \dots, D^K$. Similarly, a page or an entire book is a sequence of Q text blocks those we name regions, R_1, \dots, R_Q . Thus, a semantic unit is a sequence of text blocks possibly written across multiple pages. Such a semantic unit may start on one page and end on another. In Figure 3.5, we see an example that, using the mark-up tokens, would translate to “iI fI iM fM iM fM iF fF ...”. We want to obtain an

4. Act Segmentation and Layout Analysis

optimal sequence of semantic units for an entire book, using both the page-level context and the global context of the book. To achieve this we need to delimit each of them from $K + 1$ boundaries, β_1, \dots, β_K , using the pseudocode (4.15), explained in Section 4.1.

With the output from the previous step, H , we have the sequence of an entire book with its mark-up tokens. However, since there are no restrictions on the mark-up tokens in any of the proposed models to work at the page level, we still cannot segment the book into acts. This is because a series of inconsistencies can occur, similar to those seen in Section 4.3. Therefore, we need to apply a set of rules or constraints on that sequence.

Ideally, we would use the entire posteriorgram H to explicitly create an output sequence consistent with the CCs depicted in Figure 4.2. Let $H = H^1, \dots, H^N \equiv \vec{H}_1, \dots, \vec{H}_K$, where $K = \sum_{i=1}^n T_i$ is the total number of characters and mark-up tokens in the book. In this case, we do not select a fixed number of words. The vocabulary size, n , is the union of the different words from the reference set together with the set predicted by the model.

Since many historical books are very long, usually spanning several hundred or thousands of pages and millions of characters, K becomes excessively long, and trying to find a global optimization is prohibitive. Fortunately, many of these K elements are characters that we can afford to ignore for this type of segmentation. Clearly, only the mark-up tokens (those in S) are the ones we really need.

However, we still need to find out which vectors in H should be decoded as mark-up tokens. Therefore, we propose decimating H by deleting all vectors that are unlikely to be mark-up tokens. That is, remove from H all vectors $\vec{H}_k, 1 \leq k \leq K$ such that:

$$\arg \max_{\sigma \in \Sigma} H_{k\sigma} \notin S \quad (4.20)$$

Let $H' = \vec{H}'_1, \dots, \vec{H}'_L$ be the decimated version of H , where L is the number of vectors in H which, using Eq. (4.20), are considered likely mark-up tokens. Therefore, it is expected that L is slightly longer than the total number of images, N , but much smaller than the total number of originally decoded characters, K . Note that each tuple in H' , formed by pairs of the set S , is logically defining the Q regions we are looking for, defined in Section 4.1. In the case of using the end-to-end model with DAN, we have

the text delimited by these regions. In the other case, using the RPNs, we have, in addition to the text, the geometry of said regions.

Now, from H' , we can use the *decoding* processes proposed in Section 4.1.1 and obtain a consistent sequence of mark-up tokens.

Let $\hat{Z} = \hat{\sigma}_1, \dots, \hat{\sigma}_L$, $\hat{\sigma}_\ell \in S$, $1 \leq \ell \leq L$ be an optimal sequence of mark-up tokens decoded over H' . Then, on the original and complete posteriorgram H , let \vec{H}_j be the vector corresponding to the symbol $\hat{\sigma}_\ell$ of \hat{Z} . All components of this vector are set to zero, except for $\hat{\sigma}_\ell$; that is, $H_{j\hat{\sigma}_\ell} = 1$; $H_{j\sigma} = 0 \forall \sigma \neq \hat{\sigma}_\ell$. Finally, the rest of the vectors in H are decoded into characters like in the HTR model. This process results in a text with a series of consistent mark-up tags that delimit the semantic units. For example, for the case in Figure 3.5, the resulting text would be “i I t f I i M t f M i M t f M i F t f F ...”, where t represents a plain text piece. This example is a consistent sequence where an act starts on one page and ends three pages later, with two middle acts and a final act at the end.

As a result, we obtain a sequence of R_{q_1}, \dots, R_{q_K} regions with a consistent tagging, and therefore, K semantic units (or acts) $D = D^1, \dots, D^K$, each containing its corresponding text.

An overview of the pipeline of the whole approach is shown Figure 4.10.

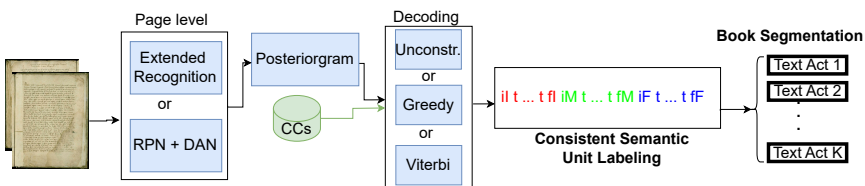


Figure 4.10: Illustration of the proposed comprehensive pipeline. The input consists of images, for which we apply one of the two proposed page-level alternatives. Then, using the CCs, we obtain a consistent sequence using one of the two alternative decoders. Finally, the acts are segmented across the book. The markup tokens are displayed in color, while text is represented as “t ... t” for simplicity.

4.4.3 Evaluation

Once we have the trained systems and the acts are segmented, we want to evaluate their performance compared to the reference partition. Since each proposal consists of several models, we can evaluate each separately if necessary.

It is important to remember that CAER simultaneously measures the transcription and segmentation of semantic units. When using an end-to-end model that performs both tasks concurrently, it becomes challenging to ascertain whether failures are due to inadequate segmentation or transcription, given that CAER does not provide a specific cause for any detected errors. However, if we employ a segmented pipeline, where detection and segmentation are performed separately from transcription, we have the capability to identify the source of errors by calculating CAER independently for each model, as we demonstrate in the results section.

Transcriptions Evaluation

The first and most direct thing we can do is evaluate the transcriptions of the HTR model, where we used DAN in both cases. For this purpose, we use the most commonly used metrics in HTR: the Character Error Rate (CER) and the Word Error Rate (WER). Using the Levenshtein distance, we define CER as the character-level difference between two text strings.

CER is defined as shown in Eq. (4.21), where \hat{y} represents the hypothesis text string and \vec{y} the reference text string. The variables s , d , and i denote the numbers of substitutions, deletions, and insertions, respectively, required to transform the reference string into the hypothesis string, employing Levenshtein distance (d). Here, n represents the total number of characters in the reference string. Moreover, since each operation (s , d , and i) incurs an identical cost, calculating CER involves summing all operations and then dividing by n , the total character count of the reference.

Similarly, WER works like CER but operates at the word level instead of the character level, using spaces to delineate words.

$$\text{CER}(\tilde{y}, \vec{y}) = \frac{s + d + i}{n} \quad (4.21)$$

It should be noted that when DAN is used at the page level, these metrics are calculated at the level of the complete page, with the reference string

of that page. Meanwhile, when working with DAN at the paragraph level, the reference string at the paragraph level is also used using the reference paragraphs. This makes the comparison not entirely fair, as one system could make a mistake when cutting paragraphs. Considering the reference paragraphs for comparison puts the page-level model at a disadvantage. However, in broad terms, this comparison is sufficient for our objective.

Detection Evaluation

On the other hand, in the second approach presented, we also have an object detection model, apart from the HTR model, at the level of these objects or regions, R_1, \dots, R_Q . Therefore, just like in [Qui22], to quantitatively measure the robustness and performance of these models and consider each layout region's alignment with its class, we use the standard COCO Object Detection metrics [Lin+15]. These are based on precision, recall, mean average precision (mAP), and Intersection over Union (IoU). Therefore, the mAP is defined as:

$$\text{mAP} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \int_0^1 \pi(r) dr \quad (4.22)$$

where r represents the recall value, $\pi(r)$ represents the precision value corresponding to r , and \mathcal{C} is the set of classes. The precision and recall values are calculated using the well-known formulas:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.23)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.24)$$

where the True Positives (TP) and False Positives (FP) values are calculated using the IoU on an alignment rule, as follows:

- FN is calculated as the number of elements in the reference set where there is no $\text{IoU}(R_i, R_j) > \text{th}, \forall R_j \in \mathcal{H}$, with \mathcal{H} being the set of hypotheses. In other words, the element R_i from the reference set cannot be aligned with any from the set of hypotheses.

4. Act Segmentation and Layout Analysis

- FP is the number of predicted elements where there is no $\text{IoU}(R_i, R_j) > \text{th}$, $\forall R_i \in \mathcal{G}$, with \mathcal{G} being the set of references. That is, the element R_j cannot be aligned with any from the set of references.
- TP is the number of predicted elements ($R_j \in H$) where $\text{IoU}(R_i, R_j) > \text{th}$ for some reference set element R_i and a threshold th , and being of the same class ($c^{R_i} = c^{R_j}$).

We compute the IoU at the pixel level between each pair of objects R_i, R_j as follows:

$$\text{IoU}(R_i, R_j) = \frac{R_i \cap R_j}{R_i \cup R_j} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FN}} \quad (4.25)$$

As we see, the mAP is calculated from a threshold (th). Then, for example, AP_{50} means that $\text{th} = 50$. On the other hand, the mAP is usually calculated by taking the average of different thresholds. In our case, we will use a range of thresholds from 50 to 95, with steps of 50 (that is, 50, 55, ..., up to 95). It is usually written as $\text{AP}_{50:90}$ or sometimes seen as $\text{AP@IoU} = 0.50 : 0.95$ [Lin+15]. In our case, we call it only mAP, as we only use this range.

However, we still need a metric that indicates how the acts have been segmented and transcribed. In other words, a metric that evaluates the system as a whole, the final result of the act segmentation and transcription. With the aim to evaluate how much information is lost due to a possible poor segmentation or alignment of the acts and the transcription of these, we use the evaluation proposed in Section 4.2, CAER.

4.4.4 Empirical Settings and Results

In this section, we present and deliberate upon the results of the act segmentation process. We examine both the detection accomplished using Region Proposal Network (RPN) models followed by transcription and the outcomes derived from the end-to-end model for simultaneous segmentation and transcription. However, before we delve into the specifics, we outline the technical aspects necessary for reproducing these results.

4.4.4.1 Empirical settings

We deployed the end-to-end DAN model, elaborately discussed in Section 4.4.1.1, adopting the same parameters as those stipulated in the original paper [CCP23]. To begin with, we trained the encoder using synthetic lines. After this, the entire model was trained for two days⁴.

Eight transformer layers were incorporated into the decoder, as suggested by the original paper. The only deviation from the original models was the omission of the last pooling operation performed by the encoder in both the Denis and Navarre corpora. This was needed because of the smaller size of the lines and the narrower inter-line spacing, which required a higher resolution to prevent information loss. The images from the Nesle and Denis corpora were converted to a resolution of 128 DPI, while those from Navarre were converted to 300 DPI.

The DAN model used for paragraph-level processing adopted the same parameters as the page-level model. However, due to the different hardware used, an early stopping criterion was set at 100 epochs based on the CER on the validation set.

A modified version of [Yim+21] was employed for both models to generate synthetic data on the fly. This process used one of the blank pages from each book as a background and superimposed French and Latin text from Wikipedia.

For the object detection models, specifically MaskRCNN, we utilized the models provided by the Detectron2 framework [Wu+19a] and the default parameters. The models were trained for 460000 steps, with a learning rate of 0.02. This learning rate was halved at the 100000 and 180000 step milestones.

4.4.4.2 Results

In Table 4.3, we present the HTR results when training DAN at the page level for the model discussed in Section 4.4.1.1, and the HTR outcomes when using DAN at the paragraph level, for the model described in Section 4.4.1.2. Generally, results tend to be superior when using DAN at the paragraph level, as it employs snippets cut from the GT. Conversely,

⁴This model was trained utilizing the v100 GPUs in the INSA cluster, access to which was provided during a research stay with the LITIS research group.

4. Act Segmentation and Layout Analysis

if we utilize DAN at the page level, the model might overlook or skip a paragraph since it first needs to be detected and transcribed.

Table 4.3: CER and WER results using the DAN model. All numbers in the table are percentages.

		DAN	DAN paragraph
Nesle	CER	5.98	5.09
	WER	18.78	17.37
Denis	CER	7.14	10.72
	WER	21.78	28.06
Navarre	CER	14.03	13.71
	WER	33.51	31.95

The Nesle corpus shows that the CER is just over 5% and the WER fluctuates between 17 – 18%. The trend with the Navarre corpus is similar, with the paragraph-level model outperforming the page-level model, albeit with a slight margin. The CER escalates to 13 – 14 points, and the WER to 31 – 33. This corpus possesses significantly more challenging handwriting than Nesle’s, making transcription considerably more complex. When considering the Denis corpus, we see that the page-level results are substantially better than those at the paragraph level. This may be attributed to overfitting at the paragraph level, preventing generalization, while the page-level model retains inter-paragraph information. The context between one paragraph and the next has likely aided in reducing error.

In the preceding table, we presented the results of DAN at the paragraph level. Given that these paragraphs need to be automatically detected at some stage, as explained in Section 4.4.1.2, we measure the outcomes of this detection using the mAP. We approach this in several ways:

- Measuring the mAP without considering the classes of acts, thus turning the problem into a paragraph detection task.
- Revealing how much mAP is lost due to poor classification rather than poor text detection, that is considering the “IMFC” classes.

- Measuring the mAP after applying the constraints (CCs) using Greedy and Viterbi algorithms.

Table 4.4: Average Precision (AP) @ IoU=0.50:0.95, without classes.

	mAP
Nesle	94.5
Denis	92.9
Navarre	90.9

In Table 4.4, we observe the results of act detection using MaskRCNN without considering classes. For this, no model has been retrained; it has been achieved simply by substituting the “IMFC” classes with a single class. We find that the mAP is greater than 90 in all cases, indicating that text detection should not present us with any problems.

Table 4.5: Mean Average Precision (mAP) @ IoU=0.50:0.95, with “IMFC” classes and before CCs (unconstrained output).

	mAP
Nesle	93.6
Denis	83.8
Navarre	76.7

We present the mAP using the “IMFC” act classes, without applying the constraints (CCs) or utilizing any decoding, in Table 4.5. We can observe that the Nesle corpus stays very close to the mAP without classes, losing only 0.9 points. This signifies that very good results have been obtained in classifying texts or detected units in the Nesle corpus, leaving little room for improvement when applying decoding algorithms later on. In the Denis corpus, where the mAP had already decreased by 1.6 points compared to Nesle without considering the “IMFC” classes, it has now dropped by 9.1 points compared to not using classes. This means the model makes mistakes or misaligns some detected hypothesis texts with references. As for the Navarre corpus, we see something similar happening. Without “IMFC” classes, it had the lowest AP, being the most challenging corpus in text detection, much like what happens with HTR. It drops from 90.9 points without classes to 76.7 points when using the “IMFC” classes, decreasing a

4. Act Segmentation and Layout Analysis

total of 14.2 mAP points. This indicates that this corpus has fewer visual clues and is more difficult to classify without considering more variables, such as the text itself or the restrictions.

Table 4.6: Mean Average Precision (mAP) @[IoU=0.50:0.95], with “IMFC” classes and after CCs.

	Method	mAP
Nesle	Greedy	93.6
	Viterbi	93.6
Denis	Greedy	85.0
	Viterbi	86.5
Navarre	Greedy	66.3
	Viterbi	83.7

The results in Table 4.5 are still inconsistent, so if we were to segment the acts, we would need to make an arbitrary decision and determine when an act that has not been assigned a clear boundary ends or begins. For instance, if two acts start consecutively, should we close one or simply not open the next one because the previous one has not yet closed? What happens if two act endings are detected in a row? Therefore, we use methods to apply the constraints (CCs): the greedy method and the Viterbi-based method. Although, as we mentioned earlier, these do not directly optimize the metrics used in this section, a strong correlation is expected, at least in most cases.

Therefore, in Table 4.6, we see the mAP after applying the CCs using both the greedy and Viterbi-based methods.

In Nesle, the results remain the same as in Table 4.5. The results from the latter table are already excellent, and there is no misclassification in any class, so these methods encounter good segmentation and do not modify it. Therefore, the point difference between Tables 4.5 and 4.4 could be due to a misalignment of some detected unit over the reference set rather than poor classification. Regarding Denis, we see that the mAP is improved compared to not using decoding, where Viterbi obtains the best results with 2.7 more points compared to not using a decoding method. Regarding Navarre, we see how the greedy method significantly worsens the results, but Viterbi significantly improves them. The most probable explanation is that the

greedy method may have caused a series of errors from the beginning, worsening the outcome. However, using the Viterbi-based method, the results improve by 7.1 points, which is also significant.

In general, using Viterbi-based decoding improves or, at least, does not worsen the results, in addition to ensuring a consistent output.

Until now, we have analyzed the results of all the systems separately. This includes the transcription level with the CER and WER, as well as the detection models with the AP. Now, we analyze the complete pipeline, both the transcription and the act segmentation, simultaneously using CAER.

Using the segmented pipeline, that is, detection models such as RPN and HTR models on said detection, we can obtain a CAER using only one part of this pipeline as the hypothesis. In this way, we can calculate the CAER using the layout obtained from Ground Truth (*layoutGT*) and the hypothesis text obtained by DAN, or vice versa, the text using the Ground Truth (*textGT*) and the hypothesis layout obtained by the RPN and finally, using both of these as the hypothesis. These results are displayed in Table 4.7, Table 4.8 and Table 4.9 for the unconstrained, greedy and Viterbi-based decodings, respectively.

Table 4.7: Results in the Nesle, Denis, and Navarre corpus using RPN to detect regions and “IMFC” classes and DAN to work at paragraph level using the unconstrained decoder. All numbers in the table are percentages.

Corpus	RPN + DAN		CAER ↓
	LayoutGT	TextGT	
Nesle	✓	X	10.85
	X	✓	0
	X	X	9.23
Denis	✓	X	19.69
	X	✓	13.27
	X	X	30.80
Navarre	✓	X	20.18
	X	✓	27.28
	X	X	44.74

In the Nesle corpus, the results in the three tables are the same. If we use the *layoutGT*, i.e., the reference layout and the hypothesis text obtained

4. Act Segmentation and Layout Analysis

Table 4.8: Results in the Nesle, Denis, and Navarre corpus using RPN to detect regions and “IMFC” classes and DAN to work at paragraph level using greedy as decoder. All numbers in the table are percentages.

Corpus	RPN + DAN		CAER ↓
	LayoutGT	TextGT	
Nesle	✓	X	10.85
	X	✓	0
	X	X	9.23
Denis	✓	X	19.69
	X	✓	7.72
	X	X	25.79
Navarre	✓	X	20.18
	X	✓	19.76
	X	X	36.36

with DAN, we get a 10.85% CAER. In other words, we lose approximately 11% of the information in the segmented acts due to the HTR model. The segmentation is perfect if we use the GT text (textGT) and the layout obtained as a hypothesis from the RPN. It is worth mentioning that this can be done because, with the RPN, we obtain coordinates for each region and with them, we can search for the text that falls within these coordinates. This has been done by calculating a threshold from an IoU. This threshold has been set at 30% for all tests; that is, if 30% of a line of text falls within the detected region, it is considered to be within that hypothesis. As we see, the RPN segments this corpus perfectly, without needing any decoding, and that is why the same results are given in both tables.

It is essential to realize that if we compare these results with the mAP obtained in the detection, we can see that, although the mAP does not reach 100 points, the CAER goes down to 0. This is because the mAP is calculated on the maximum inclusive rectangle of each act, including blank or text-free areas on the sides and upper and lower limits. In other words, if an act is detected with a large amount of spare white space, which is not in the reference, the mAP is decreased, but the detected information is the same, as the white space is not obstructive. The same does not happen if we detect extra text, for example, from another act. This is why a metric

Table 4.9: Results in the Nesle, Denis, and Navarre corpus using RPN to detect regions and “IMFC” classes and DAN to work at paragraph level using Viterbi as decoder. All numbers in the table are percentages.

Corpus	RPN + DAN		CAER ↓
	LayoutGT	TextGT	
Nesle	✓	X	10.85
	X	✓	0
	X	X	9.23
Denis	✓	X	19.69
	X	✓	8.46
	X	X	25.98
Navarre	✓	X	20.18
	X	✓	16.76
	X	X	34.72

like CAER is useful, as it allows us to compare the information between the hypothesis and the reference disregarding these details. We can also see how the CAER, with a 0% error due to segmentation, closely matches the CER obtained in the same corpus at the paragraph level, differing only by a few tenths.

In Denis, we see that using layoutGT yields a 19.69% CAER, while if we use the hypothesis layout and the reference text, we move to a 7.72% CAER using the greedy decoder and 8.46% using Viterbi. This indicates that the most challenging part and where more information is lost is in the transcription of the text, losing more than 12 points. In this case, the result with the greedy decoder is slightly better than with Viterbi by a few tenths, which does not usually happen. However, this can occur because, as mentioned earlier, Viterbi does not try to optimize the CAER but the most probable segmentation on the obtained hypothesis. Still, the difference is not significant. This difference narrows slightly using both the text and the regions detected as a hypothesis. In this case, about 26% of the information is lost in the act segmentation.

In Navarre, we see that using the reference layout but the hypothesis text loses 20.18% of the information. Using the layout obtained by an RPN and the reference text, we get a 19.76% error using the greedy algorithm and

4. Act Segmentation and Layout Analysis

Table 4.10: Results in the Nesle, Denis, and Navarre corpus using only DAN. All numbers in the table are percentages.

Corpus	Method	CAER ↓
Nesle	Unconstrained	46.91
	Greedy	58.88
	Viterbi	13.78
Denis	Unconstrained	63.32
	Greedy	41.70
	Viterbi	31.36
Navarre	Unconstrained	50.18
	Greedy	51.76
	Viterbi	41.97

a 16.76% error decoding with Viterbi, while the unconstrained decoding shows a higher error, as expected. This tells us that, in this case, there is a greater difficulty transcribing the text and not detecting it, with a 3-point difference in the best case using Viterbi. If we use both hypotheses, we get an error of 34.72 points, i.e., we lose around 35% of the information when segmenting and transcribing the notarial acts. It is evident from these results that there are still challenges to be overcome in both transcription (HTR) and layout analysis (RPN). It's noteworthy how decoding strategies can significantly influence the final results, with Viterbi usually providing more robust results.

In each corpus, different factors influence the CAER. For instance, in the Nesle corpus, the primary source of errors comes from the HTR, while the layout detection seems to work perfectly. On the other hand, in the Denis corpus, both transcription and layout detection contribute significantly to the error, although the transcription is more problematic. Lastly, both elements seem to struggle in the Navarre corpus, but the transcription proves to be particularly challenging.

On the other hand, in Table 4.10, we display the results achieved by employing DAN as an end-to-end method for obtaining transcriptions and the layout itself. This approach effectively removes the need for layout geometry labeling, accomplishing this task of the pipeline in one step.

In these results, we observe that Viterbi decoding performs significantly

better than the greedy and unconstrained decoding across all three corpora in the end-to-end approach with DAN. In the Nesle corpus, using Viterbi dramatically decreases CAER from 58.88% to 13.78%, although this is still much higher than the error rate observed when using the RPN method for layout detection.

Using Viterbi in the Denis corpus leads to a CAER of 31.36%, around six percentage points higher than when combining RPNs with DAN. Using Viterbi in the Navarre corpus leads to a CAER of 41.97%, around seven percentage points higher than the combination of RPNs with DAN. It is important to note that the segmented pipeline approach, which leverages physical layout information to train an RPN for text detection and then uses DAN for transcription, performs better than the end-to-end system. However, the comparison is not entirely fair in terms of resource utilization. The segmented pipeline requires labeled physical layout information, which the end-to-end pipeline does not need. This difference implies a higher cost for sample labeling in the segmented pipeline. On the other hand, the segmented pipeline requires fewer hardware resources, as each separate model can be trained with less powerful equipment. In contrast, the end-to-end model requires more advanced hardware. For instance, for the segmented pipeline, we used an Nvidia 2080 Ti GPU with 8GB of memory for each separate model, while the end-to-end model was trained on an Nvidia V100 GPU with 32GB of memory. The end-to-end model utilized all the memory even with a batch size of 1, making running on GPUs with smaller memory sizes unfeasible. Overall, these results highlight the trade-offs between the two approaches in terms of performance, cost of annotation, and hardware requirements. It suggests that choosing between the two approaches depends on the specific constraints and priorities of the project.

In Figure 4.11, we can observe the results of both approaches on an image from the Nesle corpus in the first step of the system, namely the end-to-end DAN model and the RPN and DAN-based system. This image consists of two pages.

Since DAN comprises transformers and its auto-regressive decoder is responsible for detecting the text, we can visualize the parts of the image that the attention model has focused on during each decoding step. The cumulative sum of each self-attention in each step results in a heatmap that

4. Act Segmentation and Layout Analysis

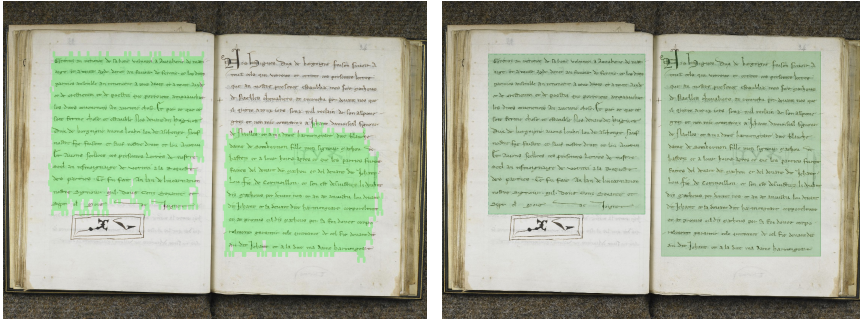


Figure 4.11: Results from a page of Nesle are presented. On the left, we see the detection executed by the DAN’s attention model, where half a paragraph remains undetected. On the right, the regions detected by the RPN on the same image are displayed.

indicates which parts of the image the attention model has attended to for transcription.

With this visualization, we can explain DAN’s behavior in terms of its visual processing. The image shown in Figure 4.11 is one in which DAN has performed poorly in the end-to-end approach, with a CER of 21%. This is evident in the image, where we can see that a significant portion of this error stems from the fact that half of the paragraph on the right page has not been detected, starting from the middle.

On the other hand, on the right side, we can see the result of the RPN model, which successfully adjusted the paragraph detection to include all the text and avoid information loss. The visualization does not include the “IMFC” labels or classes in both images.

We have another example in Figure 4.12, this time from the Denis corpus, showcasing a different type of error. In this image, an “AM” should be in the left column, and the sequence “AF, AC, AI” in the right column.

On the left, we can see the result of the end-to-end DAN model, where all the text has been detected, but the three acts in the left column have been merged into a single one. In this case, the CER has not increased compared to the average, unlike the previous case where there was undetected text. However, this type of error cannot be corrected in the second step of the pipeline, as this step allows for changing the labels of the semantic units but not cutting or merging them.

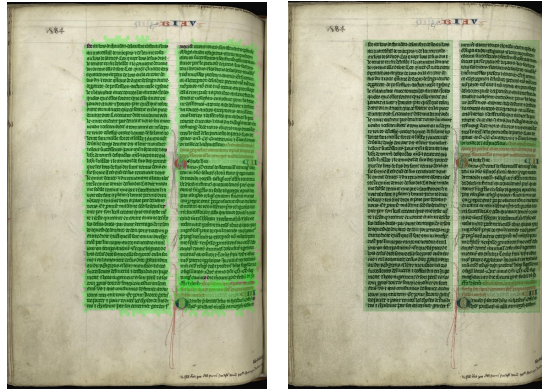


Figure 4.12: Results from a page of Denis are depicted. On the left, the detection conducted by the DAN’s attention model is shown, where a portion of a paragraph remains undetected. On the right, the regions detected by the RPN on the same image are presented.

In the right image, we can observe that the RPN has successfully detected all the text and correctly separated the sequence in the right column (the green rectangles slightly overlap, making it difficult to visualize in the image). The “IMFC” labels or classes have been omitted in the visualization.

In addition to the analysis performed on each individual model and their combinations, it is important to assess the overall performance of the entire pipeline. This includes not only the final CAER result but also the practicality of the system in terms of computational costs, ease of implementation, and robustness to changes in the data.

Let us look at how these results could guide further improvements in the system:

- **Transcription Quality:** The highest source of error in most cases is the HTR model (DAN). This suggests that efforts to improve transcription quality could lead to significant improvements. This could involve tweaking the architecture or training process of the DAN, or exploring other HTR models.
- **End-to-End vs. Segmented:** The end-to-end DAN approach provides decent results, especially with Viterbi decoding. However,

4. Act Segmentation and Layout Analysis

the segmented pipeline with RPNs for detection and DAN for transcription performs better overall. This suggests that leveraging the physical layout information to detect before transcription is a promising strategy. However, this approach is higher annotation costs that need to be considered.

- **Decoding Method:** Viterbi decoding consistently outperforms the greedy method in almost all cases. This indicates that a probabilistic approach to decoding that considers the entire sequence of detections is usually more effective than simply choosing the highest-scoring detection at each step.
- **Corpus Differences:** The performance varies across different corpora, with Nesle generally having lower error rates than Denis and Navarre. This suggests that characteristics of the individual documents, such as the quality of the handwriting or the complexity of the layout, can significantly affect the performance. This is an important consideration for developing a robust system for a wide range of documents.

In summary, while the results achieved so far are promising, several potential avenues exist for further improving the system. Carefully balancing the trade-offs between performance, computational costs, and practical considerations are crucial in these efforts.

4.5 Simancas Archive Segmentation

Until now, we have explored act segmentation across multiple distinct corpora, each with its peculiarities. In the AHPC corpus, in Section 4.3, we established the capability to segment acts, especially when faced with notable physical constraints. For instance, the necessity for acts to commence and/or conclude on a singular page ensures no two acts coexist on the same page. Subsequently, we moved on to a more intricate act segmentation, in Section 4.4, without these restrictions. We created an architecture enriched with additional elements and incorporated act transcription. This revamped approach was then evaluated across three different corpora.

In this section, we employ a new corpus to examine both scenarios and the dual methodologies for tackling the issue — with or devoid of

constraints. We evaluate the outcomes of each approach using the same metric. Specifically, our analysis draws from the Simancas Archive corpus, previously elaborated upon in Appendix A.4.

However, before starting, we confront an additional challenge with this corpus. As delineated in Appendix A.4, our collection comprises groups of pages with acts. Yet, not every act seamlessly begins and concludes within our labeled pages. This scenario suggests a potential encounter with an act initiating on a prior, *untagged* page or an act commencing on our page but not concluding within our labeled set of pages. Consequently, we must revise the *Markov Chain*, shifting to the model presented in Figure 4.13 and apply it for every group of pages independently. This modified Markov Chain denotes the flexibility to initiate at any act segment and culminate at any subsequent segment within a page group. Still, all preceding rules should be followed after the group’s initiation, except at the end. This adaptation also necessitates a shift in our methodology: instead of examining an entire book, our focus narrows to individual page groups, treating each as an isolated book entity. Also, some change is needed in the boundary pseudocode, where now we are forced to set a boundary at the end of the group of pages, as follows:

```

 $b_0 := k := 0;$ 
for ( $j := 1 \dots Q$ ) if ( $c_j = F \parallel c_j = C \parallel c_j = Q$ ) { $k := k + 1; b_k := j$ };  $K := k$ 

```

4.5.1 Applying Page Restriction

In Section 4.4, we assumed a series of constraints that appeared in the AHPC corpus. Namely, two acts could not coexist on the same page, and each act had to span at least two pages. For the Simancas corpus, we intend to replicate the tests from the AHPC, excluding the second condition, which negated the presence of acts labeled as “Complete” or “C”.

To emulate this environment, we segment parts of the acts from the Simancas corpus, treating each segment as an individual page. Consequently, each piece of the act is now a separate image. This implies that, rather than working with the G complete images, we transition to utilizing Q images. Each of these images is treated as an individual page, consistent

4. Act Segmentation and Layout Analysis

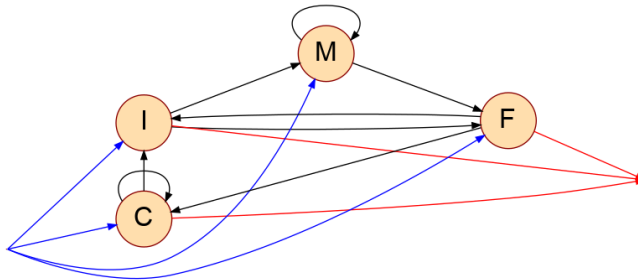


Figure 4.13: Markov Chain representing Consistency Constraints for “IMFC” tagging for the Simancas Archive. This Markov Chain is applied for every group of pages.

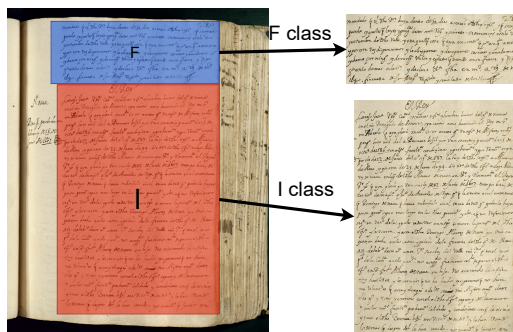


Figure 4.14: Parts of the act from the Simancas archive cut out.

with the approach previously employed in the AHPC corpus. This bypasses the need to detect acts within pages since we operate as if this information is already provided. This is illustrated in Figure 4.14.

As delineated in Table A.5, we have a compilation of 291 regions (now, images) allocated for training, encompassing both complete and incomplete acts. For the testing phase, the total number is 99, of which 52 are complete acts and 47 are incomplete.

We follow the same evaluation methods as with the AHPC corpus, shown in Section 4.3.3. We also continue with the same configuration shown in Section 4.3.4. However, only the best-performing model from that section, ResNet50 with DP as the decoder, is used.

Table 4.11: Report of the classification errors without consistency-enforcing decoding for the Simancas Archive acts, jointly with the CAER using PD decoding.

Classifier	Err.	(%)	W.Err. (%)	CAER	CAER + HTR
ResNet50	50	41.32	50.18	16.08	49.35

4.5.1.1 Results

In this section, we delve into the segmentation outcomes of the Simancas Archive. This analysis assumes that we accurately detected the segments of acts. As a result, our focus is on classification without needing detection, coupled with dynamic programming for a precise and consistent segmentation, as elaborated in Section 4.3.2. Regarding the CAER, our initial approach leverages the reference text. This approach aims to specify the purest segmentation error, devoid of influences from transcription or text detection errors. Subsequently, we employ the hypothesis text derived from HTR techniques. This helps us discern the error proportion attributable solely to segmentation, sidelining HTR-related inaccuracies.

It is worth highlighting that in this particular corpus, the HTR facet of the Simancas Search project is not encompassed within this thesis’s scope. In this thesis section, the primary emphasis is on achieving act segmentation. Still, we present the results incorporating the HTR derived from the Simancas Search project, elucidating the extent of retrievable information from combined segmentation and transcription efforts. In contrast to earlier sections, instead of utilizing DAN, we have adopted PyLaiia[PM18], a toolkit rooted in Convolutional Recurrent Neural Networks (CRNN) focusing on line-level operations. The line extraction was facilitated by the P2PaLA software[Qui17], which is underpinned by a UNet design optimized for image segmentation, followed by a connected component algorithm for ensuing line detection and extraction. The resulting CER is 17.2% while the WER stands at 35.3%.

Table 4.11 shows us the CAER errors when classifying the regions of the reference deeds with GT text in the first column, as well as using the hypothesis text in the second column.

Initial observations indicate a 16.08% consistent act segmentation error,

4. Act Segmentation and Layout Analysis

excluding HTR deviations. It appears to have adeptly segmented most of the acts, even in the face of classification challenges stemming from maximum likelihood evaluations. Such efficiency signals the robust calibration of probabilities by the ResNet50 neural network, assuring that decoding primarily zeroes in on the most probable trajectories, ultimately leading to precise and consistent act segmentations.

It's important to highlight, as outlined in the corpus description at Appendix A.4, that we are not dealing with the entire book but rather with distinct groups of pages. Applying individual decoding to each page group might positively impact the results. Setting precise boundaries between acts, it could serve as a safeguard, preventing segmentation errors from permeating across different page groups. In the last column, we see the CAER for act segmentation, derived using the hypothesis text provided by HTR. This error approaches nearly 50%, which was somewhat anticipated given the inherent error observed in the HTR results.

In the next step, we implemented the methods outlined in Section 4.4. While utilizing the same HTR, we trained an RPN with ResNet50 as the backbone. This allowed us to identify the regions associated with the acts and apply a decoding method. Now, we no longer operate under the previous assumptions and transition to detecting and classifying acts without referencing the labels, which are only used to train the system. Therefore, in this case, we again have G images, which are the original pages, and Q regions to detect.

In Table 4.12, we observe the mAP for detection using RPNs. Initially, without considering the classes of detected regions, we registered an mAP of 51.85. However, when accounting for the four classes without decoding, the mAP drops by 10 points. Once decoding is employed, there's a slight decline, settling the mAP at 41.17.

Interestingly, one might anticipate an mAP boost post-decoding. Yet, this has not materialized here. A potential reason could be that, while the region's classes may alter during decoding, they don't necessarily shift towards the correct class. We are essentially tracing the most likely pathway with decoding but with certain inherent constraints, which might inadvertently introduce labeling errors or exacerbate existing ones. As the table shows, this process might have negatively impacted classification or AP. However, the CAER might still show improvement. This is because CAER

Table 4.12: Mean Average Precision (mAP) @ IoU=0.50:0.95, without classes and with classes after the decoding process using DP in the Simancas Corpus.

	AP
Without classes	51.85
IMFC without decoding	41.80
IMFC after DP decoding	41.17

factors in the textual content of the acts, whereas mAP primarily focuses on the region. Consequently, an extensive region with little text and incorrectly labeled content could dramatically reduce the CAER value. Conversely, the degradation by the same mislabeled region in segmentation error may be less pronounced, given its little textual content.

Table 4.13 presents the segmentation error, specifically the CAER. For a comprehensive analysis, we integrated techniques from the preceding sections. Specifically, we compared using solely the RPN for detection (as showcased in the table’s final two columns) with a strategy where acts were detected without class assignment and subsequently classified using a pre-trained act classifier from the last experiments. From the first column, employing the RPN exclusively for detection combined with a distinct classifier and utilizing the reference text, we achieved a CAER of 32.62%. Transitioning to the HTR text led to a CAER increase, reaching 59.35%. Conversely, when we used only the RPN for both detection and classification purposes, and with the reference text, the CAER descended to 18.23%. This signifies a nearly halved error relative to the combined RPN and classifier approach. In our conclusive observation, employing the RPN alongside the HTR text resulted in a CAER of 50.37%. This configuration outperformed the combined RPN and classifier approach by a margin of 9%.

In summarizing this section, we have successfully lowered the CAER to 18% on this notably challenging corpus. Yet, when we adopt the automatically derived hypothesis text, the error spikes to 50%. This suggests that half of our act segmentations could be erroneous, which is too high to be deemed “useful”. While there’s room for enhancement in pure segmentation, with a potential reduction of 18 points, the predominant source of

4. Act Segmentation and Layout Analysis

Table 4.13: CAER (%) for the Simancas Corpus after using Viterbi decoding.

	RPN with Classifier		Only RPN	
	Text GT	HTR	Text GT	HTR
CAER	32.62	59.35	18.23	50.37

inaccuracies emanates once the HTR text is integrated. Consequently, any efforts to refine these results should prioritize enhancing the HTR, albeit this lies beyond the scope of this thesis.

4.6 Discussion

We have attempted to address the issue of act segmentation by approaching the problem from two distinct perspectives. Firstly, we adopted an approach predicated on some significant assumptions; secondly, we employed a perspective assuming fewer restrictions or assumptions about the data. Both methodologies yield a variety of differing conclusions.

In Section 4.3 we presented a comprehensive examination of the application of deep learning classifiers and decoding techniques, specifically the Viterbi algorithm, for the classification and segmentation of books at the act level with some hard assumptions like a minimum size of an act and each act has to start in different pages, evaluating the models in this framework with the CAER. While our methodology yields promising results, some areas could potentially enhance the performance and applicability of the developed system.

A major finding in this study is the performance of ResNet50. Despite other classifiers, ResNet50 emerged as the best-performing model with the lowest CAER when used with Viterbi decoding. It shows the ability of mid-sized networks like ResNet50 to capture the necessary feature space for this classification task without overfitting that might be encountered by larger networks. Notwithstanding, it is worth acknowledging the instances where misclassifications occurred. Through a detailed examination of the misclassified images, we gained insight into the complexity of the problem. Errors seemed to be mainly due to specific characteristics in the images, such as a large portion of a page being blank or the presence of signatures. These

findings could inform future model development and training strategies, where such factors are given additional consideration.

Interestingly, one of the intriguing directions we found during this study is the need for more utilization of contextual information or metadata associated with the books, which is not used in our approach. Incorporating such data may significantly enhance the system’s ability to perform classifications and segmentations accurately, leading to a potential area for future research. Moreover, our study focused on two specific books. While this provided a reasonable sample size for this preliminary research, future work could benefit from including a larger and more diverse selection of books. Such an extension could reveal additional challenges and complexities in book segmentation, thereby developing more robust and generalizable systems. Overall, this study represents a first step in automating the book segmentation process at the act level, which holds significant implications for the information extraction from historical books and documents.

Next, in Section 4.4 we presented a system to perform the act segmentation without the hard assumptions explained before; that is, with now a minimum size of notarial acts, and each act can start and end on the same page, adding much more difficulty to the problem.

The model we have employed diverges from the previously proposed one. Given that we can no longer rely solely on image classification but must delve deeper within each image, we have opted for RPNs over an image classifier. Nevertheless, the backbone for feature extraction in the RPN is still a ResNet50. We could have also used the RPN to classify images in the AHPC corpus under the assumptions outlined, treating each image as a singular bounding box to be consistently detected. However, having a pure image classifier with the same backbone, we believe, is more direct, faster, and considerably less costly, yielding potentially very similar results.

Additionally, we have transcribed the HOME corpus, which was not feasible with the AHPC corpus because we do not have the GT transcriptions required to train a system. Simultaneously, we utilized the DAN model at the full-page level as an alternative to the RPN, but not in the AHPC corpus because we need the transcriptions to train it.

Following the experiments without the hard assumptions in Section 4.4, our experiments’ results have given us valuable insights and indications

4. Act Segmentation and Layout Analysis

of how to further improve the segmenting and transcribing of notarial acts. One of the most significant results is the consistent performance improvement achieved by combining the RPN and DAN models compared to the end-to-end DAN model. This segmented approach, which separates the text detection and transcription process, performs better in the CAER across all tested corpora. However, this advantage comes with a price: the segmented approach requires the availability of the physical layout information. On the other hand, it requires less resources in terms of hardware. This could be a crucial factor when considering the scalability and applicability of the model, especially when deploying it in real-world scenarios where resource limitations may be a significant constraint. As for the transcription performance, our results indicated that the HTR model, namely the DAN, was the main source of error in most cases. This suggests that improving the transcription model could significantly impact the pipeline's overall performance. Several avenues can be explored to achieve this goal, such as exploring other HTR models or fine-tuning the existing model to better suit the nature of the data. The choice of decoding method was another significant factor affecting the performance. The Viterbi decoding method usually outperformed the greedy method in our experiments. This suggests that considering the entire sequence of detections rather than making independent decisions for each step could provide more accurate results. Also, the difference in performance across various corpora illustrates the complexity and variability of handwritten texts. The model performance was significantly affected by the individual characteristics of the documents, such as the handwriting quality and the layout's complexity. This underlines the importance of developing robust and adaptable models for a wide range of documents to ensure reliable performance in real-world applications.

Our final analysis re-examined the most effective techniques using the Simancas Archive corpus. We employed both classification techniques and a combination of detection and classification, also testing the combination of both. Our findings underscore that RPNs excel when tasked with simultaneous detection and classification rather than segmenting acts through separate processes.

In conclusion, this section has presented two distinct pipelines for different notarial act segmentation scenarios. Moreover, in the second case, we have proposed two options in terms of neural network models for the

first stage, thereby dividing the initial step into two segments. Thanks to this approach, we can opt to segment acts in various situations. For instance, we might have assumptions discussed in Section 4.3, where a lighter model for page classification might suffice. Variations of this model could also be introduced, such as having entire acts confined to a single page (i.e., eliminating the restriction on the number of pages per act, $M(k) \geq 2$), which would, in turn, alter the corresponding Markov model.

Next, suppose we have physically labeled notarial acts without the hard assumptions but not the transcriptions of the pages. In that case, we can train the model explained in Section 4.4 based on RPNs, without utilizing any Handwriting Text Recognition (HTR) model. Since the text is not used to classify paragraphs into C classes at this stage, it would not be necessary for segmentation (although this would block the measurement with the CAER).

Finally, if we have the transcription, we can use the RPN model alongside an HTR model or employ an end-to-end model such as DAN to perform the entire first stage of the pipeline in a single step.

Document Classification

5

Vast quantities of digital reproductions of historically significant manuscripts are diligently safeguarded in archives and libraries worldwide. A notable portion of these materials captures the quotidian activities of bygone eras. Our interest lies mainly in those, as mentioned earlier in the last sections, historical notarial deeds, an extensive category of archived documents that provides a rich narrative of our past. Typically, individual deeds in these series are clustered into sizeable collections, housed within boxes or larger bundles, each potentially comprising hundreds of deeds and thousands of page images. The sheer magnitude of such document collections often impedes the provision of comprehensive metadata to accurately encapsulate the content of each bundle and each distinct deed. We use the term bundle as in the last chapter, which may encompass several, and often numerous, “image documents”, alternatively referred to as “files”, “acts”, or “deeds”, the latter specifically about notarial image documents discussed herein. We already segmented these documents into acts or deeds in the last chapter. However, these are presumed to fall under various categories or classes, which perhaps furnish the most crucial information in describing a manuscript. In this chapter, our central focus is on a task termed Content-Based Image Document Classification (CBIDC). The primary goal is to classify an untranslated image document into a predetermined set of classes or types, which could span from a handful to several hundred pages of handwritten text. These classes or types correspond to the topics or semantic content expressed by the text within the images. Existing techniques for content-based document classification are predicated on the assumption that documents consist of electronic text, where characters, words, and paragraphs are explicitly presented. Therefore, the prevailing method for tackling the CBIDC task would entail the initial transcription of the images, followed by standard document classification techniques. However,

5. Document Classification

manual transcription is practically unfeasible, while attaining sufficiently accurate automated transcripts often proves elusive for extensive collections of historical manuscripts.

It’s vital to discern that the CBIDC task under discussion here distinguishes itself markedly from other similar-sounding but functionally distinct tasks. Examples include “document classification” (DC, as previously discussed, applicable only to unequivocal electronic text), “content-based image classification” (pertains to single images of natural landscapes, not text), and “document image classification” (where classifications relate to the visual aesthetic or page layout of single images).

Furthermore, it’s crucial to note that recent strides in document classification, inclusive of those employing multimodal approaches and visual transformers [SOE22; Xu+21], are ill-suited to our CBIDC task. The nature and scale of textual visual objects contemplated here (potentially hundreds of page images) differ significantly and are sizably larger in comparison to the single-image objects considered in these studies. In [SOE22], images are resized to 1170×827 and in [Xu+21] images are resized to 224×224 . In both works, they process one image at a time. In historical handwritten images, we usually need a much higher resolution to save some details from the text to recognize it. Also, we process many images at a time using their textual content. Trying to process many images at a higher resolution with computer vision techniques would be very expensive in terms of computation.

Additionally, it is imperative to acknowledge that document types evolve over time. In a realistic setting, we must contend with image documents representing classes that have not been previously encountered. Within the conventional classification schema, all such novel image documents would invariably be misclassified. Consequently, to proficiently manage the proposed task, it is necessary to identify new image documents that don’t belong to any existing class; in other words, the system should decline or “reject” their classification. A pivotal contribution of this study is to candidly address this comprehensive CBIDC challenge and propose efficacious solutions within the OSC framework, explained in Section 2.3.1.

The initial challenge we face is demonstrating the effectiveness of our classification method within a conventional closed setting, “CSC”, in contrast to transcriptions in the traditional sense. Once this has been

accomplished and the methodology’s viability confirmed, we employ the “OSC” framework for classification with rejection. We use the AHPC corpus previously used in the preceding section to segment notarial acts for all these steps. As a result, this classification is the natural progression of the notarial act segmentation task; once the acts have been segmented, we aim to determine the class of each one.

Unlike deed segmentation, as detailed in Appendix A.2.2, we limit our scope to only the JMBD4949 and JMBD4950 bundles. This restriction is because these are the sole collections for which we possess GT for each class in every deed.

5.1 Problem Definition

We begin by defining the problem of “CSC” in its most classical form within the Pattern Recognition (PR) perspective. Here, we have a document, which could consist of one or more text images, X , hailing from the collection \mathcal{X} . We postulate that each document belongs to one of the C *known classes*, with the class set closed. Each document X is represented in a vector form, \vec{X} . Under the minimum statistical error framework, an optimal prediction of the class of the document X is by following Eq. (2.21) [DH+73].

The posterior probabilities $P(c \mid \vec{X})$ can be calculated in several ways. For instance, using Support Vector Machines, Multinomial Naive Bayes (MNB), or a Multilayer Perceptron (MLP). The input to each of these systems is always \vec{X} , and the output is a posterior probability, $P(c \mid \vec{X})$, $1 \leq c \leq C$, for each class c .

The most common method for evaluating a CSC classifier is through its error probability, estimated by the *Error Rate* k_e/K . Here, k_e represents the number of incorrect predictions made on a test set of K image documents from the same C classes considered during training [DH+73].

In the CSC framework, all the classes to be recognized are always known. However, in real-world scenarios, this is often not the case. Frequently, a set of known classes and awareness of other unknown classes exist. For instance, when classifying notarial acts, we have a set of classes or typologies well represented in the bundles we use. However, we know other typologies exist in other bundles that we have not seen and are not

5. Document Classification

represented in the set used for training. This is where the CSC paradigm ceases to be as helpful, and we can shift our focus to the OSC (Open Set Classification) framework, where we assume several classes, $\tilde{C} > C$, will exist in the collection \mathcal{X} .

As we have hinted in Section 2.3.1, a preliminary approach to OSC, from the CSC perspective, is to train with the C *known classes* and an additional class, the *rejection class*, which would comprise the remaining $\tilde{C} - C$ unknown classes. Adding this extra class would necessitate *unknown class* data for training. However, we could create this class using underrepresented classes (for example, those with too few training samples). Using the *rejection class* would give us $C' = C + 1$ classes [DH+73], and the minimum error-risk classification would still be achieved using Eq. (2.21), replacing C with C' . Moreover, we could still use the *Error Rate* to evaluate the system.

However, there are other ways to tackle this problem without using unknown class data, i.e., using only the C known classes. This is done by using a threshold t on the class posteriors to determine when a document from the reference set should be rejected, belonging to a *reject* class, which would not be used for training. This is implemented using Eq. (2.22).

This can be addressed with the model used previously (for instance, an MLP giving the posterior per class). Nonetheless, we can also adapt other ideas that have tried to solve the OSC problem, albeit for different tasks and architectures entirely different from those we propose for our problem.

Shu et al. [SXL17] propose a model called “one versus rest” (1-vs-rest) in which they use a neural network with an output configured as a vector of C activation functions with *sigmoids*. Then, each output c corresponds to a Bernoulli distribution, $P(b_c | \vec{X})$, $1 \leq c \leq C$, where b_c is the value of a random boolean variable, where if the class c of X is 1 and 0 otherwise. In this case, we used an MLP architecture. We replaced the SoftMax output layer, which corresponds to the categorical distribution $P(c | \vec{X})$, with a 1-vs-rest layer and used binary cross-entropy (following Eq. (2.9)) as the loss function, as in [SXL17]. In the future, we refer to this model as a “binary-outputs MLP” (bMLP).

On the other hand, Yang et al. [Yan+22a] propose a prototype-based model, OSC. An input stack of convolutions is used for feature extraction from the input image. In our CBIDC task, the input does not consist of a

single image but multiple ones (from a few to hundreds), represented vectorially by their textual content. The original model presented in [Yan+22a] would find it very challenging (or impossible) to handle such a large amount of data in the original (image) format.

We follow their formulation but instead of a CNN as feature extractor we use an MLP.

Also, in [Yan+22a] they experiment with two loss functions. A discriminative loss than can be Distance-based cross-entropy loss or a loss “One Versus All” or OVA loss function, similar to the used in [SXL17]. In our case, we refer to as pMLP when using the DCE loss. When using the OVA loss function, which we already referred to as 1-vs-rest, we refer to it as pbMLP.

When using the OVA loss function, which we already referred to as 1-vs-rest, we refer to as pbMLP.

5.1.1 OSC Thresholds

Defining a threshold t - either fixed or somehow estimated - all the methods proposed so far, such as bMLP and pbMLP, can be implemented within the OSC framework with REJECT, following Eq. (2.22) and assuming that $P(c | \vec{X}), 1 \leq c \leq C$ are the output probabilities given by the model in question. Similarly, the Error Rate for OSC can be calculated in the same way.

In [SXL17], some heuristics are proposed to calculate different thresholds, one per class. However, we have implemented them, and it seems that in our specific problem, no improvement is achieved compared to having a single threshold for all classes, so we continue using a single threshold for simplicity and clarity. Therefore, we have considered using two simple heuristics to calculate this threshold.

The first one is proposed in [SXL17], where it is calculated as $t = 1 - \sqrt{\sum_X (1 - P_{\hat{c}}(X))^2} / K$, where K corresponds to the total number of samples with known classes, $P_{\hat{c}}(X)$ is calculated using the model-dependent probability ($P_{\hat{c}}(X) = P(\hat{c}(X) | X)$ for MLP or $P_{\hat{c}}(X) = P(b_{\hat{c}(X)} | X)$ for bMLP), and $\hat{c}(X)$ is the correct class of X according to the GT.

The second way we propose to calculate the threshold is by calculating the mean of the maximum posteriors in the test set samples. This can be

5. Document Classification

done since we do not use the class provided by the reference set, but we use the sample's posterior obtained by the model.

It is worth mentioning that this value could also be adjusted by the user, as depending on the system and the input data (unknown in a real environment), the user may want one behavior or another from the system. However, to evaluate the system's performance with these thresholds, we show the ROC curve [MRS08] for all possible thresholds. The area under this curve, called AUROC, is typically a scalar that adequately measures the overall system performance for all rejection thresholds. The ROC curve assumes a binary decision, so in our case, we decide whether the sample X is or is not within the C known classes.

5.2 Feature Selection and Extraction for CBIDC

As we alluded to in the previous section, we require a vectorial representation of each image's text. To surmount the hurdle of dealing with historical images that have not been transcribed, we shall employ probabilistic indices (PrIx), as previously explained in Section 3.2.1. PrIx have already proven their utility when dealing with collections with a high level of uncertainty in transcription hypotheses due to poor image conditions, extremely challenging handwriting, and an array of issues that can emerge from historical documents. By using PrIx, we can *estimate* textual features. Given that R is a boolean random variable, we can regard the relevance probability (RP) $P(R \mid x, v)$ - where the (pseudo-)word v is written in x - as a statistical expectation.

Within this framework, we can estimate all the probabilities and frequencies needed for vector representation. We base this on, firstly, a selection of the (pseudo-)words with the highest IG (refer to Eq. (2.2)), and a vector representation grounded in Tf-Idf (see Section 2.1.2.1).

To estimate and acquire the expected number of words written in x , all the RPs of all the pseudo-words indexed in an image x are summed. From this principle, we can estimate the remaining probabilities and frequencies in the following manner.

We have $n(x)$, the total number of words written in an image x , and $n(X)$, the total number of words written in a document X , typically comprising several pages. $n(v, X)$ is the frequency of a specific word v in X .

And with $m(v, \mathcal{X})$ being the number of documents from a collection \mathcal{X} that contain the word v , we can compute the expected values as follows:

$$\begin{aligned} E[n(x)] &= \sum_v P(R | x, v), \quad E[n(v, X)] = \sum_{x \subseteq X} P(R | x, v) & (5.1) \\ E[n(X)] &= \sum_{x \subseteq X} E[n(x)], \quad E[m(v, \mathcal{X})] = \sum_{X \subseteq \mathcal{X}} \max_{x \in X} P(R | x, v) \end{aligned}$$

With this in place, and given that M is the total number of documents in \mathcal{X} , we can calculate the Tf·Idf value in the following manner:

$$\text{Tf·Idf}(v, X) = \frac{E[n(v, X)]}{E[n(X)]} \cdot \log \frac{M}{E[m(v, \mathcal{X})]} \quad (5.2)$$

Therefore, in our proposed approach, a document X is represented as a feature vector $\vec{X} \in \mathbb{R}^n$, indexed by n words from a vocabulary V_N , where $\forall v \in V_N, X_v = \text{Tf·Idf}(v, X)$.

We have just mentioned that we use a closed vocabulary of n words, which means we need to select those words that provide the most pertinent information for the problem we seek to resolve. We have chosen *Information Gain* (IG) to achieve this. As detailed in Section 2.1.2.2, we need to calculate a range of probabilities for each v , which we can estimate using the statistical expectations in Eqs.(5.1). Following Eqs.(2.3), $m(v, \mathcal{D})$ is estimated as $E[m(v, \mathcal{X})]$ and $m(v, \mathcal{D}_c)$ is estimated $E[m(v, \mathcal{D}_c)]$, resulting as follows:

$$\begin{aligned} P(t_v) &= \frac{E[m(v, \mathcal{X})]}{M}, & P(c | t_v) &= \frac{E[m(v, \mathcal{X}_c)]}{E[m(v, \mathcal{X})]} & (5.3) \\ P(\bar{t}_v) &= 1 - P(t_v), & P(c | \bar{t}_v) &= \frac{M_c - E[m(v, \mathcal{X}_c)]}{M - E[m(v, \mathcal{X})]} \end{aligned}$$

where a subset of documents \mathcal{X}_c from \mathcal{X} belongs to class c . M_c represents the number of documents in \mathcal{X}_c .

5.3 Open Set Classification in AHPC

In this section, we demonstrate that our proposal, using PrIx to estimate probabilities, calculating IG for all pseudo-words, and their Tf·Idf, work

5. Document Classification

within the CSC framework in a closed, CSC problem, classifying the acts of the AHPC corpus. However, as previously mentioned, there are very few problems in the real world where the number of classes to be classified is fixed. Therefore, we classify also within the OSC framework.

As shown in Table A.3, the closed set of classes is $C = 12$. However, at least another 29 classes are known to exist. This makes the total number of known classes, whether in training or testing, $\tilde{C} = 12 + 29 = 41$ in total, reducing the proportion of genuinely known classes to 29.3%, and treating the rest as unknown classes (although, as previously explained, we know the class of some acts, but they are not sufficiently represented, and we treat them as unknown classes). In the AHPC corpus, we classify 498 documents within the 12 known classes and another 57 documents, which, ideally, should be rejected using the OSC framework (or classified within the reject class, if applicable).

In [GHC21], the concept of *openness* is defined and calculated for \mathcal{X} as:

$$O(\mathcal{X}) = 1 - \sqrt{2C/(C + \tilde{C})} \quad (5.4)$$

where C is the number of known classes and \tilde{C} is the total number of classes, i.e., the number of known classes plus the number of unknown classes. The *openness* indicator equals 0 when the problem is entirely closed (as in CSC, i.e., $\tilde{C} = C$) and higher when there are more *open*, i.e., unknown and outside the training set, classes. In AHPC, the openness is $1 - \sqrt{2 \cdot 12 / (12 + 41)} = 0.327$. The openness serves us to compare how open a problem is against future corpora, although beyond the scope of this thesis. It should also be noted that often the openness of a problem is not known for sure, but a minimum openness bound can be considered, at least, since we may only know some of the open classes but a lot of them. The usual thing would be for new classes to appear and the openness to increase.

5.3.1 Experimental settings and Results

In this subsection, we first present the results using methods without any threshold, such as the CSC framework and the OSC but with a reject class. Subsequently, we use threshold-based methods to reject unseen classes and

show the results already using the OSC framework. However, before all, we first discuss the empirical settings to reproduce these experiments.

5.3.1.1 Empirical Settings

Since the PrIx vocabulary is typically large due to many low-probability hypothesis pseudo-words, it has been deemed necessary to limit it. Hence, we eliminate all pseudo-words with an RP lower than 0.1 ($P(R | x, v) < 0.1$). This reduces the original vocabulary from 809 787 pseudo-words to 55 927. Subsequently, words were sorted by IG, and a vocabulary V_n of n words was chosen exponentially between 16 and 16 384. Finally, an n -dimensional Tf-Idf vector was created for each document to be classified.

We consider three MLP configurations with a different number of hidden layers, which all are followed by batch normalization and a ReLU activation function. Once again, the most basic configuration is a basic multilayer perceptron with $C = 12$ outputs or $C' = 13$ outputs when using a reject class. We call this model MLP-0, as it has zero hidden layers. The following configuration is MLP-1, consisting of a hidden layer of 128 neurons. Finally, we have MLP-2, pMLP-2, bMLP-2, and bpMLP-2 with two hidden layers of 128 neurons in each model. Deeper models have been tested but with worse or equal results.

The parameters of each MLP, bMLP, pMLP, and bpMLP have been initialized following [GB10] and trained using cross-entropy loss for MLP and pMLP or binary cross-entropy loss for bMLP and bpMLP for a minimum of 20 epochs and a maximum of 500, using early stopping with a patience factor of 50 epochs. For the MLP-0, the RMSprop optimizer with a learning rate of 0.1 has been used, while for the rest of the models, SGD [Rud17] with a learning rate of 0.01 has been used. Also, following the recommendations of [Yan+22a], a single prototype per class has been used in pMLP and bpMLP, and after several tests carried out, the best size for said prototype is 128.

We consider all the acts available in both files (JMBD4949 and JMBD4950) as a single dataset. This means that we have 498 documents spread over 12 classes for CSC and 555 documents spread over 41 classes for OSC. In Table 5.1, we can see a quick reminder of the classification data for AHPC, shown in Appendix A.2.2.

5. Document Classification

Table 5.1: Number of documents and RWs for JMBD4949 and JMBD4950.

Class ID's	Deeds	Estimated RWs
PA	240	859.3
LP	72	1112.3
DB	44	1310.9
LE	32	1441.1
TE	29	2279.3
SA	21	5902.5
RI	17	1356.9
CS	12	2901.2
DP	10	874.5
ST	9	716.7
CN	6	1335.9
TF	6	1404.5
REJECT	57	2369.9

We follow the *leaving one out* protocol to make the partitions. This means we can have specific problems when calculating IG and Tf·Idf, as it can be costly to calculate it each time for each test (the 498 or 555 times). However, we have observed that leaving a sample or keeping it within the set (the test one) does not significantly affect the calculations, and we have chosen to carry out these calculations, as well as for the thresholds, with all the samples simultaneously. Regarding the threshold, we know that this simplification somewhat breaks the principle of the independent test set; it should be noted that the values of these estimates are not critical, as will be discussed later.

5.3.1.2 Threshold-less Closed and Open Set Classification

The results in Figure 5.1 unfold in alignment with Eq. (2.21). We discuss the conventional outcomes derived from CSC, deploying three MLP models. These models are trained and evaluated with a narrowed scope, relying solely on samples from the 12 known classes. Subsequently, we turn our attention to OSC results, leveraging the same model framework but with an expanded dataset comprising 13 classes. This extension includes the 12 known classes in conjunction with a distinct REJECT class, which contains

samples from an additional 29 classes. The results, illustrated following the escalating dimensionality (the number of words selected by IG) of the Tf·Idf image document embeddings, offer a comprehensive view of both classification methods.

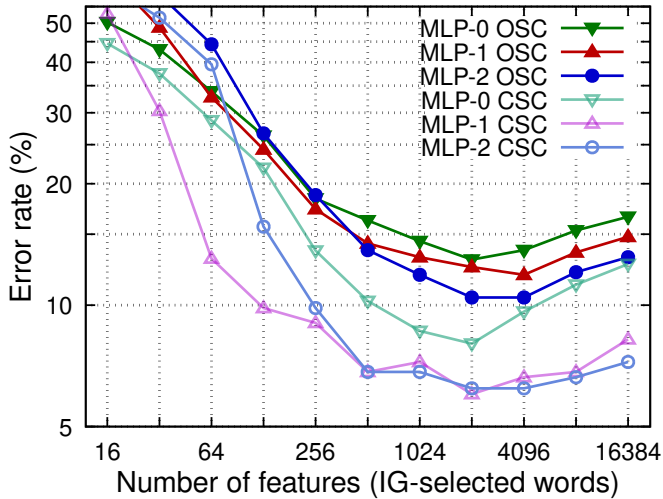


Figure 5.1: Leaving-one-out classification error rate on JMBD4949 and JMBD4950 with three threshold-less MLP models, both for Closed and Open Set Classification. OSC: training and testing with 12 known classes; OSC: training and testing with 12 known plus REJECT (13 “classes”). All the results are based on PrIx document and word frequency estimates. 95% confidence intervals (not shown for clarity) are all smaller than $\pm 4.4\%$ and smaller than $\pm 3.0\%$ for all the error rates below 15%.

As was expected, CSC outcomes surpass their OSC counterparts. Under the traditional CSC paradigm, the results infer that MLP-1 and a minimum of 512 words for Tf·Idf representation can successfully classify more than 93% of our image documents, or “deeds”, with their correct class. The MLP-2 model presents the most efficacious OSC and CSC results when the image documents are embedded within a 2048-dimensional Tf·Idf vector space. These results are shortly summarised in the first column of Table 5.2.

Table 5.2 also hosts comparable results from the bMLP-2 classifier. Despite its output layer and training loss not pursuing the maximization of class discrimination, this model performs on par with MLP-2. The precision

5. Document Classification

Table 5.2: Classification error rate of threshold-less methods. CSC: training and testing with 12 known classes; OSC: training and testing with 12 known plus a REJECT “class”. Results are shown for $n = 2048$ words and both PrIx image representations and plain text HTR image transcripts.

Classifier	PrIx				HTR
	MLP-2	bMLP-2	pMLP-2	pbMLP-2	MLP-2
CSC ($C = 12$)	6.2	6.2	7.0	11.7	8.0
OSC ($C' = 13$)	10.5	11.0	10.8	18.2	12.3

of the classification achieved is noteworthy, considering the intricate nature of the task: to classify sets of untranscribed manuscript images with up to 12 (or $12 + 1$) subtly different classes defined by nuanced word combinations.

Outcomes for the prototype network models denoted as MLP-PN - specifically, pMLP-2 and pbMLP-2, as discussed in Section 5.1, are also included in this table. For the pure CSC model (12 classes), the outcomes of pMLP-2 align with those of MLP-2 and bMLP-2, yet the precision of the pbMLP-2 model, which was trained similarly to bMLP-2, lags behind noticeably. Outcomes for these models trained with the additional REJECT class (OSC, $C' = 13$) exhibit a trend akin to all other models, albeit pbMLP-2’s precision falls short.

For completeness, Table 5.2 also features results procured with the identical MLP-2 classifier. However, here the state-of-the-art HTR image transcripts [Sán+19; Rom+19b], rather than PrIx, are deployed for image representation. Here, the documents and word frequencies requisite for IG and Tf·Idf were computed naively, starting from the noisy plain-text HTR output. As anticipated, these results do not match those procured with our proposed approach, wherein document and word frequencies are estimated using PrIx image representations instead of direct computation.

Finally, Table 5.3 discloses the confusion matrix and the error rate per class for MLP-2 OSC. It notes that the REJECT class is implicated in 38 out of the 58 errors.

These outcomes underscore the prowess of the MLP models, particularly MLP-2, in classifying image documents within both closed and open set contexts. The bMLP-2 model, despite its lack of the maximization of class discrimination, rivals MLP-2’s performance, indicating its potential viability as an alternative for such tasks.

Table 5.3: Confusion matrix for PrIx MLP-2 OSC with $n = 2048$.

		JMBD4949 & JMBD4950														Total	Err (%)
		PA	LP	DB	LE	TE	SA	RI	CS	DP	ST	CN	TF	RJ			
PA		229	0	0	0	1	0	0	2	0	0	1	0	7	240	4.6	
LP		2	66	2	0	0	1	0	0	0	0	0	0	1	72	8.3	
DB		3	1	37	0	0	0	0	0	0	0	0	0	3	44	15.9	
LE		1	1	0	29	0	0	0	0	0	0	0	0	1	32	9.4	
TE		1	0	0	0	27	0	0	0	0	0	0	0	1	29	6.9	
SA		0	0	0	0	0	19	0	0	0	0	0	1	1	21	9.5	
RI		0	0	0	0	0	0	17	0	0	0	0	0	0	17	0.0	
CS		0	0	0	0	0	0	0	8	0	0	0	0	4	12	33.3	
DP		0	0	0	0	0	0	0	0	10	0	0	0	0	10	0.0	
ST		0	0	0	0	2	0	0	0	0	5	0	0	2	9	44.4	
CN		0	0	0	0	0	1	0	0	0	0	5	0	0	6	16.7	
TF		0	0	0	0	0	0	0	0	0	0	0	6	0	6	0.0	
REJECT		3	6	3	0	0	2	0	3	0	1	0	0	39	57	31.6	
Total		239	74	42	29	30	23	17	13	10	6	6	7	59	555	10.5	

The prototype-based models, MLP-PN, present promising results for closed set classification but grapple with difficulties once a REJECT class is introduced in an open set context. This suggests potential challenges in accommodating the novelty and uncertainty of unknown classes. Overall, we have observed that introducing a REJECT class invariably escalates the error rate, underlining the augmented challenge that the incorporation of unknown classes presents in classification tasks.

Using Header for Classification

As observed in Appendix A.2.2, a significant portion of the relevant information is often contained in the rectangles located at the upper-left corner of each deed starting page. Given this, it is worthwhile to test the best-performing model within the CSC framework to see how it fares in classifying these specific areas and then compare the results with those obtained from the full text of the deeds. Figure 5.2 presents the results of using only the PrIx found in these header rectangles against the best results showed in Figure 5.1 using the MLP-2 model.

We observe that when utilizing a limited number of features –up to 64– by relying solely on the PrIx from the headers, we achieve an error rate of 24%. The line terminates at this point because the headers do not contain

5. Document Classification

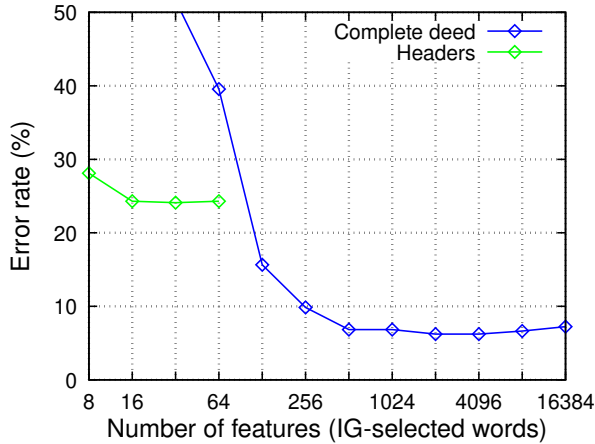


Figure 5.2: Leaving-one-out classification error rate on JMBD4949 and JMBD4950 with MLP-2 and Closed Set Classification solely using PrIx from headers or the complete deed. All the results are based on PrIx document and word frequency estimates. 95% confidence intervals (not shown for clarity) are all smaller than $\pm 4.4\%$ and smaller than $\pm 3.0\%$ for all the error rates below 15%.

more than 64 running words, and the line would otherwise remain constant. However, as we increase the feature count to 128 or more, it becomes evident that leveraging the entire deed for classification yields superior results. Consequently, we have decided to discard using only headers for classification purposes and continue to employ the complete deeds. Using this approach within the CSC framework, we have achieved an error rate of approximately 6%.

5.3.1.3 Threshold-based Open Set Classification and Rejection

In the presented scenario, our models undergo training exclusively on samples from a set of 12 known classes. Nonetheless, the test set introduces an additional layer of complexity as it encompasses samples from not just these known 12 classes but from an additional 29 classes considered unknown. This culminates in a task demanding both classification and rejection capabilities.

The OSC error rates encapsulated within Table 5.4, elucidate the duality of this task. Echoing the section prior, these rates capture a triad of error types: conventional misclassification within the known classes, wrongful rejection of samples from known classes, and the failure to reject samples emerging from unknown classes.

Table 5.4: OSC classification + rejection bMLP-2 error rate for different thresholds (t), using PrIx and $n=2048$ words with the bMLP-2 model. It was trained with $C = 12$ classes and tested with samples of all $\tilde{C} = 41$ classes (12 known, plus 29 REJECT “classes”). 95% confidence intervals are within $\pm 3.2\%$, or $\pm 2.2\%$ for the lowest error rate.

Threshold estimate	bMLP-2	(t)
Fixed 0.0	15.9	(0.00)
Fixed 0.5	16.4	(0.50)
$1 - \sigma$ [SXL17]	6.5	(0.75)
Avg. max class posterior	7.2	(0.94)
Best on test (“oracle”)	6.5	(0.75)

Without a trained REJECT class, our OSC must comply with Eq. (2.22), necessitating a threshold t . Table 5.4 reveals results for a pair of fixed thresholds and a further pair of thresholds estimated in line with the guidelines laid out in Section 5.1.1. A supplementary *oracle threshold* is also incorporated, established as the threshold that elicited the lowest error rate within the test set.

Four models detailed in Table 5.2 were subjected to this comprehensive, threshold-dependent OSC scenario. The OSC error rates attained under the guidance of the Oracle threshold were as follows: MLP-2: 13.0%, bMLP-2: 6.5%, pMLP-2: 16.57%, and pbMLP-2: 18.37%. Due to the evident superiority of bMLP-2, detailed results for this model alone are displayed in Table 5.4.

The results achieved with the two estimated thresholds are similar and close to optimal. For bMLP-2, the exact estimates are not critical as comparable error rates were observed across the entire threshold spectrum ranging from 0.70 to 0.97.

Overall, we can infer that bMLP-2 exhibits remarkable accuracy in a comprehensive, threshold-dependent OSC environment. Its performance

5. Document Classification

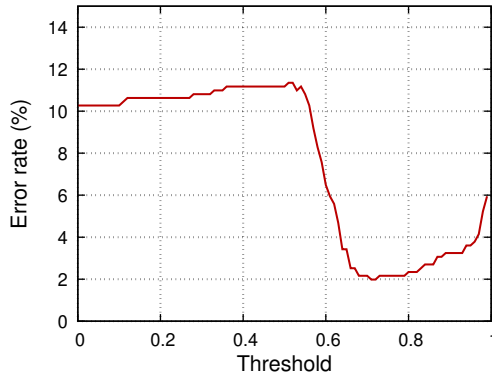


Figure 5.3: Rejection performance for bMLP-2 OSC with PrIx and $n=2048$ words. Training with $C = 12$ classes, testing with samples of all $\tilde{C} = 41$ classes. Rejection error rate (%) for a range of thresholds t and a resulting AUROC value of 96.68%.

parallels the best results secured in a basic CSC framework, with the added responsibility of rejecting samples from unknown classes.

Figure 5.3 depicts the Error Rates for binary classification (the “Reject” vs. “non-Reject” scenario) across a threshold range from 0 to 1 when utilizing the bMLP-OSC model with 2048 words. The resulting AUROC is 96.68%, succinctly summarizing the substantial rejection potential across the entire range. The rejection performance manifested by bMLP-2 approaches perfection, accounting for the previously discussed OSC superiority of that model.

Several key conclusions can be drawn from all these results shown in the section. It is noteworthy that models can be efficaciously trained solely on samples from known classes and can still demonstrate aptitude when confronted with a test set encompassing samples from unknown classes, adeptly managing the classification and rejection of samples.

Upon examination of the trialed models, bMLP-2 stands out due to its exceptional performance, delivering impressive classification accuracy and an equally competent rejection performance for the unknown classes.

In addition, applying estimated thresholds has proven to be an effective strategy for regulating the rejection rate in OSC. These estimates can deliver results bordering on the optimal, though their precision is not critical for

certain models. For instance, bMLP-2 showcased comparable error rates across a broad threshold range.

Lastly, threshold-based OSC has demonstrated itself as an effective strategy. It offers results comparable to those secured in CSC, with the added advantage of being able to handle the rejection of samples from unknown classes.

5.4 Discussion

Emerging from the findings elucidated in this chapter, we can distill several significant insights.

The initial pivotal aspect of our discussion orbits around the efficacy of PrIx and the proposed pipeline, consisting on the estimation of all probabilities explained in Sec Section 5.1 using PrIx for calculate and ordenate the words by IG and create a vectorial representation using Tf·Idf, in environments where reference transcriptions are absent, and HTR error is moderately high. This PrIx pipeline can effectively navigate these challenging environments, offering a viable solution for accurate classification. Models leveraging PrIx have outperformed in both CSC and OSC scenarios, demonstrating their robustness and adaptability.

The improvements of the PrIx-based pipeline can be explained by its prowess in efficaciously grappling with uncertainty. Contrary to traditional models that operate optimally under ideal conditions, those harnessing PrIx can sustain their performance in more realistic and complex scenarios characterized by elevated noise levels and ambiguity, as exemplified by deteriorating ancient handwritten documents. This discovery implies that PrIx could be an invaluable asset in a plethora of practical applications and not just confined to classification, where acquiring reference transcriptions is challenging or high HTR error rates are anticipated.


The second salient observation from our analysis pertains to the utility of the OSC environment in tackling real-world problems that encompass an open set of classes. While most traditional classification tasks presuppose a closed set of classes, many real-world problems involve scenarios where unknown or unforeseen classes may surface. Our findings indicate that OSC could be a potent approach to managing such problems.

5. Document Classification

Our analysis, particularly in the OSC scenario hinged on calculating a threshold t , revealed that OSC models, especially the bMLP-2 model, can provide high accuracy rates. Remarkably, this high performance was sustained even when the models were tasked with rejecting samples from unknown classes - a challenge often faced by traditional models.

The employment of threshold strategies in OSC was found to be effective. These strategies supplied a method for managing the rejection of samples from unknown classes. Moreover, it was shown that threshold-based OSC yields results on par with those of the basic CSC, but with the added capacity to handle unknown classes.

Information Extraction in Structured Documents



Information extraction in ancient structured documents has seen a rising interest in recent years due to the necessity to digitize and access the wealth of information these documents harbor. This endeavor proves especially challenging due to the variability in document structure and formatting and the occurrence of handwritten text and specialized vocabulary.

Old structured documents often comprise tables and continuous text that carry invaluable information. For instance, historical documents might feature tables cataloging daily weather conditions, detailing data such as sea temperature, atmospheric pressure, and wind direction; all logged onto a preprinted template. Such tables can teem with abbreviations, numerical information, citations, and other artifacts scattered across cells. Furthermore, each table page might be linked to an accompanying page of descriptive text elucidating the same day in plain text. We can see several exemplars of tables displaying these features in Figure A.5.

Information extraction from these documents is laden with several challenges. In specific cells, the anticipated information might be replaced by quotation marks, indicating that the data expected in a cell is identical to that from a preceding row in the same column. These quotation marks pose multiple difficulties. From a layout analysis perspective, these marks are hard to detect due to their significantly small size. From an information extraction viewpoint, these marks pertain to relevant information in previous rows, necessitating the identification of the marks and the information they reference.

Other complexities associated with these types of pages, previously mentioned in Appendix A.3, include text inscribed between cells, cells with information dispersed over several lines, struck-through column names, numbers represented as superscripts, or the same information recorded in varied manners. Moreover, the absence of context among the cells further

6. Information Extraction in Structured Documents

complicates the recognition process, as language models typically employed after subsequent HTR processes are less advantageous than in continuous text or outside the table.

Existing solutions addressing the recognition of table images primarily rely on machine learning techniques [RS20]. To make progress in processing images of historical tables via machine learning methods, it is essential to prepare databases with their corresponding GT. This preparation is a laborious and time-consuming manual task, resulting in a scarcity of historical corpora concerning information extraction in handwritten tables.

In summary, extracting information from old structured documents is a challenging yet crucial field for digitizing and accessing historical information. Despite significant strides in this area, there is substantial room for improvement, with further research needed to overcome existing challenges. Hence, we propose a pipeline segmented into various phases for Information Extraction (IE) in structured documents. Due to the associated challenges, we utilize the HisClima corpus (see Appendix A.3), as this corpus offers us a series of interesting difficulties that no other historical corpus, as far as we know, offers us.

6.1 Problem Definition

In a given image X , our objective is to identify and extract structured data, typically organized in tables. These tables are partitioned into distinct information cells, denoted as v , which may or may not contain text. Cells are categorized into two types: value cells and header cells. Value cells generally hold the information we aim to extract, while header cells serve as indices to locate these value cells within the table. For instance, if we search for “Winds Direction Magnetic at 3”, the term “Winds Direction Magnetic” would constitute a column header cell, and “at 3” would be a row header cell. We refer to this combination of header cells as a “query”. The corresponding value cell is found at the intersection of these header cells within the table; in this example, it would be “Eaxv”. This process is illustrated in Figure 6.1. Thus, the problem boils down to identifying the triplets (v_c, v_r, v_v) that form the table for each image.

Our approach addresses the problem by initially identifying the text within the image. For instance, this could involve detecting lines in the

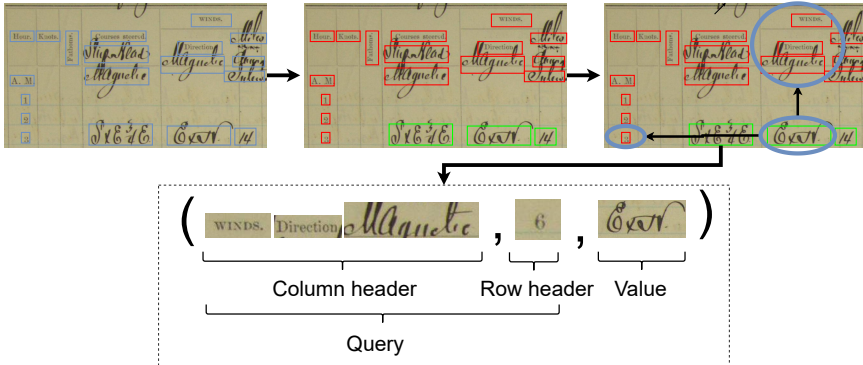


Figure 6.1: In the first step, an image segment featuring a table is displayed, with the text already identified. Subsequently, the detected regions are categorized into header and value regions in red and green, respectively. In the final step, a query is exemplified by selecting one of the value regions, searching for its column header and row header, and extracting this data as a triplet.

image, although the method is flexible and could be adapted to detect other elements, such as individual words or entire cells. We define these elements or regions of interest as t each characterized by a specific geometry $\vec{r} \in \mathbb{R}^4$ and classified into one of three categories $t_c \in \{v, r, c\}$, depending on whether it is a value cell ($t_c = v$), a row header ($t_c = r$), or a column header ($t_c = c$).

A cell, denoted as v , is composed of one or more regions v , and the classes of these constituent regions determine its classification. For instance, a value cell, represented as v_v , consists of a sequence of N regions, each of which is of the value type. Mathematically, this can be expressed as $v_v = t^1, \dots, t^N$, where $\forall t_c = v$.

Once the regions of interest are detected, the next step is establishing relationships between them to form triplets and extract the relevant information. In a triplet (v_c, v_r, v_v) , the value cell v_v and the column header v_c belong to the same column. Likewise, the value cell v_v and the row header v_r are part of the same row. By identifying the substructures (i.e., rows and columns in the context of tables) to which each cell belongs, we can determine the column headers for each table column and the row headers for each existing row. Furthermore, we can intersect the rows and columns

6. Information Extraction in Structured Documents

to form the triplets by identifying the value cells—those that are not header cells. This process is illustrated in Figure 6.2.

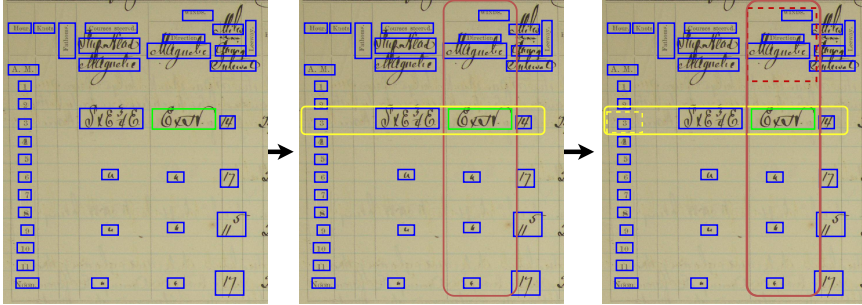


Figure 6.2: In the illustration, we start with a given table and focus on a pre-selected random cell, highlighted in green. We then identify the row and column this cell belongs to, marked in yellow and red, respectively. After determining these two substructures, their respective headers are indicated with dotted lines.

To address these challenges, we propose using *GNNs* for node classification and substructure detection. Each region of interest in this framework is a node in the graph, characterized by its geometric attributes. We initiate the process with a preliminary graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The objective is to identify an optimal adjacency matrix \mathcal{A}^* that encapsulates the graph of the targeted substructures. Since the adjacency matrix \mathcal{A} can be derived from the edge set \mathcal{E} and vice versa, our focus is on determining \mathcal{E}^* to construct \mathcal{A}^* , which represents the new graph based on the initial graph \mathcal{G} . The search for \mathcal{A}^* can be conducted as follows:

$$\begin{aligned}
 \mathcal{A}^* &= \arg \max_{\mathcal{A}} P(\mathcal{A} \mid \mathcal{G}) \\
 &= \arg \max_{\mathcal{E}} \prod_{e_{ij}^* \in \mathcal{E}^*} P(e_{11}^*, \dots, e_{ij}^* \mid \mathcal{G}) \\
 &\approx \arg \max_{\mathcal{E}} \prod_{e_{ij}^* \in \mathcal{E}^*} P(e_{ij}^* \mid \mathcal{G})
 \end{aligned} \tag{6.1}$$

where e_{ij}^* is an element of \mathcal{E}^* iff $P(e_{ij}^* \mid \mathcal{G}) \geq t_G$. We can derive \mathcal{A}^* assuming that the edge probabilities are independent.

We obtain the probability for each edge $e_{ij} \in \mathcal{E}$ as to whether that edge belongs to \mathcal{A}^* . We describe it as

$$P(e_{ij}^* | \mathcal{G}) = P(Z = z_{ij} | \mathcal{G}, i, j) \quad (6.2)$$

where Z is a random binary variable that assumes the value $z_{ij} = 1$ if the edge e_{ij} should be included in \mathcal{A}^* and 0 otherwise. An edge e_{ij} should be in the adjacency matrix \mathcal{A}^* when it connects a region of interest to another within the same substructure—specifically, within the same row when identifying rows or within the same column when identifying columns.

The Eq. (6.2) is estimated using a *GNN* that incorporates an MLP for its output. This MLP, composed of the same number of layers and neurons per layer as the GNN, followed by batch normalization and ReLU activations, has a sigmoid activation function for each edge e_{ij} in the graph in the last layer. The inputs to the MLP are $|\vec{x}'_i - \vec{x}'_j|$ and \mathcal{T}_{ij} , where \vec{x}'_i and \vec{x}'_j represent the node embeddings computed by the GNN, and \mathcal{T}_{ij} denotes the pre-calculated, invariant edge features derived from the initial graph. The MLP's weights are trained in conjunction with the GNN, as a single neural network architecture, and the training uses binary cross-entropy as the loss function for all MLP outputs.

Once the GNN is trained, it yields the probability for each edge $P(e_{ij}^* | \mathcal{G})$. This results in the creation of a matrix \mathcal{A}^* , which belongs to the set $\{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$, defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } P(e_{ij}^* | \mathcal{G}) \geq t_G \\ 0 & \text{else} \end{cases} \quad (6.3)$$

where t_G serves as a threshold value, which is commonly set at 0.5.

Finally, the connected components within \mathcal{A}^* are extracted, where each component serves as a distinct substructure. This process is illustrated in Figure 6.3 using a small table and a simple initial graph. The graph is pruned in two distinct ways to isolate rows and columns, with each identified substructure highlighted in a different color.

A GNN is employed to classify the regions of interest as either value cells or header cells. Given that these regions are represented as nodes in the graph \mathcal{G} , the task essentially becomes one of node classification. In a manner

6. Information Extraction in Structured Documents

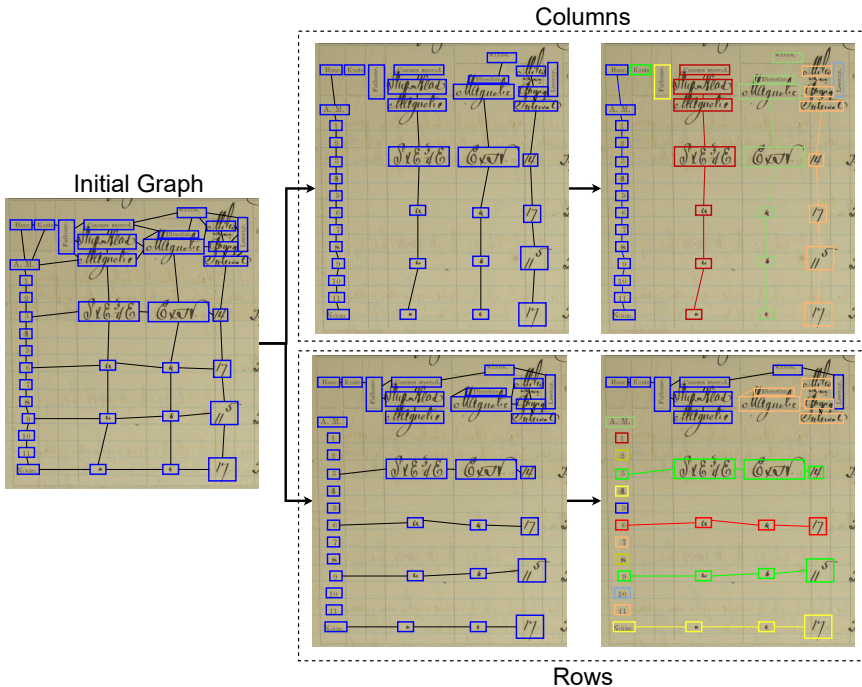


Figure 6.3: This illustration demonstrates how rows and columns are identified from a simple initial table graph. In the top row of the illustration, the graph’s edges are pruned to isolate regions that belong exclusively to the same column. Subsequently, a Connected Components algorithm is applied, and each column is color-coded differently. The process for identifying rows is similarly depicted in the bottom row of the illustration.

analogous to the edge classification, we derive a probability indicating whether each node in the graph serves as a header node, $P(C = c_i | \mathcal{G}, i)$:

It is worth noting that our cells may consist of multiple regions, commonly seen as multi-line cells. The approach we propose is capable of handling this complexity. By intersecting a row and a column, we can identify a complete cell, regardless of whether it spans multiple lines or not.

Another challenge we aim to address is the issue of “multi-span” cells. This situation appears when a column splits into two, sharing a portion of the column header, thus causing both columns to have a common query term. To deal with this, we propose using a directed graph that focuses solely on cells identified as headers. The graph’s directionality corresponds

to the reading order. When two regions of interest from different cells are connected in this graph, it indicates a “multi-span” cell. This initial graph is significantly smaller than the one used for other substructures due to fewer nodes. The network is trained similarly to the others but with a slight modification: the absolute value used in the graph embeddings is removed, making the input to the MLP $\bar{x}'_i - \bar{x}'_j$. This ensures that the value for the tuples (\bar{x}'_i, \bar{x}'_j) differs from (\bar{x}'_j, \bar{x}'_i) . After edge pruning with the probabilities output from GNN, we start from each 'node with no outgoing edges' in the resulting graph and follow the path to the main node of each substructure, thereby identifying each *multi-span cell*. This process is illustrated in Figure 6.4, which starts from a graph consisting only of regions identified as column headers.

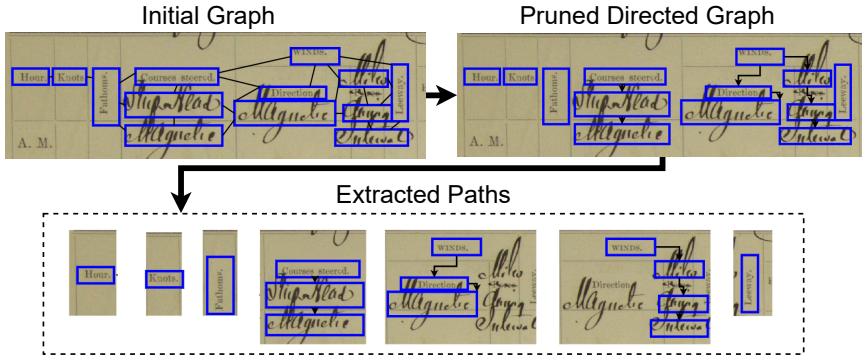


Figure 6.4: An example of how an initial undirected graph formed only by header regions is converted into a pruned directed graph in the first step. In the second and final step, all the paths that form the header *queries* are extracted from each node without outgoing edges.

Once we reach this point, we have mapped the entire table structure. The next step is transcribing the detected regions of interest using some HTR tool. Finally, we classify the header cells v_c based on their textual content following:

$$\hat{y}_c = \arg \max_{y_c} P(t | Y = v_c)P(Y = Y_c) \tag{6.4}$$

where t represents the textual contents of the header cell v_c , and the random variable Y takes on values from a set of attributes that we aim to extract,

6. Information Extraction in Structured Documents

depending on the specific problem at hand. To estimate the conditional probability $P(t \mid Y = v_c)$, we employ a character-level n-gram language model for each attribute y_c . The prior probability $P(Y = Y_c)$ is directly estimated using maximum likelihood methods.

In Figure 6.5, we provide a schematic representation that outlines the various components and the overall pipeline for the information extraction methodology proposed in this section.

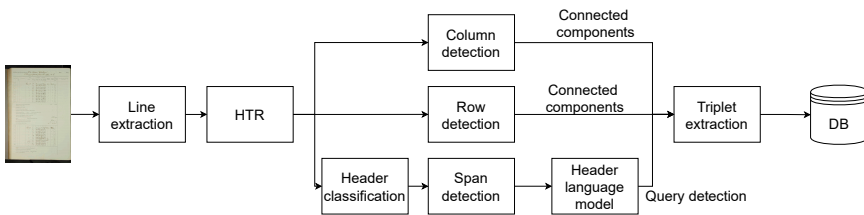


Figure 6.5: A comprehensive overview of the end-to-end pipeline designed for information extraction.

It is important to emphasize that while the methodology has been specifically tailored to extract information from handwritten tables, its applicability extends far beyond this narrow scope. The approach can be adapted to handle a wide range of structured documents, including various table formats, forms, and documents that rely on a “key-value” data structure. Using a specific table type as a case study simplifies the explanation, but the method’s versatility should be noticed.

6.2 Evaluation Measures

As the preceding section outlines, our information extraction pipeline consists of multiple specialized models, each addressing a different part of the problem. Consequently, it is crucial to evaluate the performance of each model in addition to assessing the pipeline as a whole. This comprehensive approach enables us to identify our system’s strengths and weaknesses.

The first component of our information extraction pipeline is text detection, specifically focusing on identifying lines within the image. To evaluate the effectiveness of this line detection, we employ the harmonic mean (F_1) between the P-value and the R-value, as outlined for baseline detection in the study by [Grü+17]. The R-value quantifies the proportion of correctly

identified baselines in our model relative to a reference set. At the same time, the P-value accounts for segmentation errors after aligning the detected baselines with the reference baselines. Given our specific application and reliance on these baselines, we find it sufficient to report solely the harmonic mean, denoted as F_1 .

$$F_1 = \frac{2PR}{P + R} \quad (6.5)$$

where P represents P-value and R stands for the R-value.

We direct the reader to the study by [Grü+17] for an in-depth explanation and implementation details. In this research, we employ the authors' original implementation, which is publicly available¹. It is important to clarify that our approach detects text lines using a neural network built on RPNs, rather than directly identifying the baselines. Nevertheless, extracting these baselines from the detected text lines is straightforward and can be achieved using dynamic programming algorithms, allowing us to maintain the use of this evaluation metric.

To train an HTR model, we use the reference lines. Consequently, it is crucial to assess the model's performance. For this purpose, we employ CER and WER metrics, specifically at the line level, using the reference lines for evaluation. The CER is defined as the Levenshtein edit distance between the reference and hypothesis text strings, denoted as \vec{y} and $\vec{\hat{y}}$, respectively. This distance measures the minimum number of character-level substitutions, deletions, and insertions required to transform the transcribed text into the reference text. The CER is then normalized by dividing it by the total number of characters in the reference text, following Eq. (4.21).

Similarly, the WER is calculated the same way as the CER but operates at the word level rather than the character level. In this case, spaces are used to delineate individual words.

Next, we employ various GNNs to extract the table structures. We use metrics that focus on the graph-based results to measure their performance. Specifically, we use the *classification error rate* for classifying header nodes or textlines. Similarly, this metric is applied to classify edges in the directed graph for detecting multi-span structures, especially since we expect a limited number of nodes and edges when focusing solely on table headers. For

¹<https://github.com/Transkribus/TranskribusBaseLineEvaluationScheme>

evaluating the detection of table sub-structures, such as rows and columns, we adopt the partition evaluation method proposed in [PDM19a]. Once these sub-structures are identified, we have a collection of *sets* representing each row and column. Utilizing the *Intersection over Union* (IoU) metric as a measure of similarity and the *Hungarian* algorithm [Kuh55] for alignment, we align each detected structure with its corresponding reference structure. A structure is considered correctly detected if it achieves a 100% match with its aligned reference. For instance, let us say the reference sets of textlines, identified by their IDs, are $\hat{Y} = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8\}\}$. In our hypothesis, we have $Y = \{\{1, 2, 3\}, \{4, 5, 6, 7, 8\}\}$. We would align the first set in each list since they are a 100% match. However, the remaining two sets would be treated as a substitution and a deletion because one set is missing and the other is incomplete. After establishing this alignment, we compute an structural error rate, denoted as F_1 , following Eq.(6.5).

Finally, to assess the overall performance of the information extraction pipeline, we use the F_1 score. This involves comparing the triplets extracted by the complete pipeline against a reference list of triplets. A triplet is considered a *True Positive* (TP) if it matches a reference triplet on the same page without repetition. Otherwise, it is considered a *False Positive* (FP). Any triplet not found in the reference list is considered a *False Negative* (FN). Using these TP, FP, and FN values, we calculate precision, recall, and, ultimately, the F_1 score, which is reported. We also compute 95% confidence intervals ($\alpha = 0.025$) using the bootstrap method with 10 000 repetitions.

6.3 Information Extraction in HisClima Tables

This section systematically explores and presents the results at each stage of our proposed pipeline, offering a detailed discussion for each. Ultimately, we disclose the final results and conduct a comprehensive analysis of the strengths and weaknesses of our approach.

Both the transcriptions and the models have been trained and tested on two datasets –Jeannette and Albatross– simultaneously. We have performed tests for methods based on GNNs using each dataset individually and a combined dataset (J+A). This approach allows us to assess the model’s ability to generalize across different image types and to understand the

impact of having a larger training dataset. Additionally, we have employed IoU to label graphs based on automatically detected lines. This serves as an estimation for metrics as if they were based on a reference graph. While this labeling is an approximation and may contain inaccuracies, it provides valuable insights into the model’s performance. The final IE results serve as the definitive measure of the model’s effectiveness.

It is important to highlight that the node features in all GNN-based models remain consistent and pre-calculated in the initial graph. The features include the coordinates of the textline’s upper-left and lower-right points, both of which are normalized relative to the overall dimensions of the image. Additional attributes encompass the textline’s height and width, the total count of pixels overlapped by neighboring elements in both horizontal and vertical directions, and the number of such neighboring elements in each orientation.

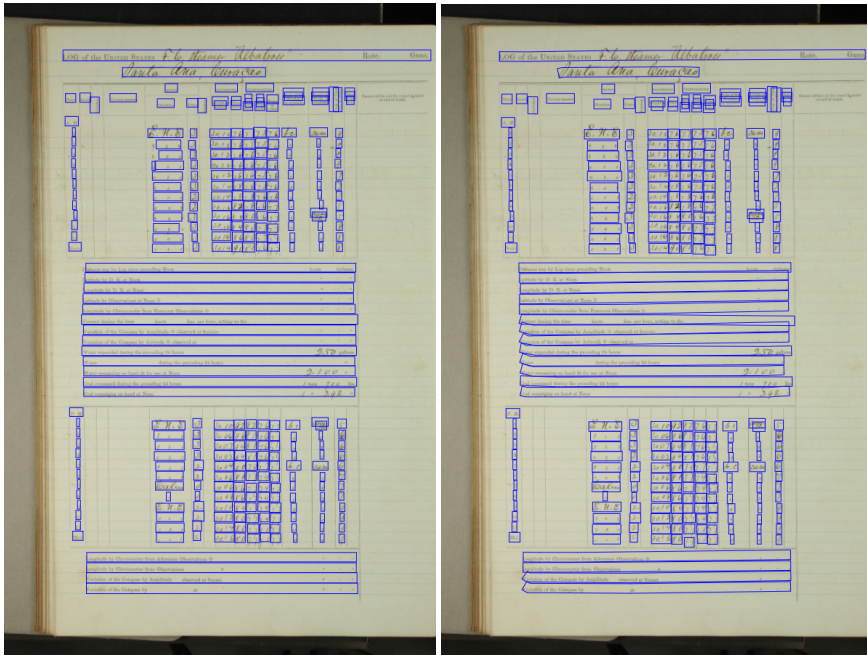
6.3.1 Textline Detection and Transcription

For text detection, we utilized MaskRCNN, treating each textline on the page as a distinct region. We employed a ResNet-50 model pre-trained on ImageNet as the backbone for MaskRCNN. Given the high volume of regions that needed to be detected, it was essential to fine-tune certain parameters of the RPN. Specifically, we set the number of objects for detection pre *non-maximum-suppression* at 5 000 and post *non-maximum-suppression* at 2 000. Without these modifications and relying solely on the default settings of the Detectron2 toolkit, the model detected far fewer textlines, resulting in suboptimal performance. The model is trained for 280 000 iterations. As a result, we achieved an F_1 score for the textline detection of 93.0. Figure 6.6 provides an illustrative example of line detection, allowing for a comparison with the reference. While the model detected nearly all lines, there were minor inaccuracies in the shape of some lines, potentially omitting some text, as seen in some lines in the fifth column of the first table.

It is important to emphasize that this step is crucial, as if a textline is not detected, it is impossible to “recover” that lost information later.

For the HTR task, we employed PyLaia [PM18], a model that combines convolutional neural networks (CNNs) with recurrent neural networks (RNNs). Specifically, our architecture features five convolutional layers with feature maps of sizes 16, 32, 48, 64, and 64, each utilizing 3×3

6. Information Extraction in Structured Documents



(a) Detected textlines.

(b) Reference textlines.

Figure 6.6: Automatically detected textlines at the left with MaskRCNN and reference textlines at the right.

kernels. We opted for LeakyReLU as the activation function and did not apply any image reduction techniques. Following the convolutional layers, the network includes recurrent layers of 128 BiLSTM neurons each.

Upon training the model, we integrated a 10-gram character-level language model generated directly from the training transcriptions using the SRILM toolkit [Sto02]. This language model was applied uniformly to both printed and handwritten text. For subsequent models in our pipeline, we use the 1-best transcription as the input.

Regarding text recognition, the performance metrics presented in Table 6.1 showcase the results on test pages across different datasets. These figures are based on lines identified in the GT, ensuring that all labeled lines contribute to the calculated values. Overall, the error rates – encompassing both printed and handwritten text – are quite low. Specifically, for the combined datasets (J+A), we achieved a CER of 2.60% and a WER of

Table 6.1: Results of text recognition for Jeannette (J) and Albatross (A). M, P and O refers to manuscript, printed and overall text respectively. 95% confidence intervals are never larger than 0.9% for manuscript text, 0.4% for printed text and 0.3 % overall. All numbers are percentages.

Corpus Test type	Jeannette			Albatross			J+A		
	M	P	O	M	P	O	M	P	O
CER	4.14	1.21	1.72	14.19	1.48	5.56	6.92	1.27	2.60
WER	6.82	1.60	3.45	18.83	1.92	10.48	11.20	1.67	5.39

5.39%. However, it is crucial to differentiate between printed text (P) errors and those associated with handwritten text (M). As expected, transcribing handwritten text is a more intricate challenge, leading to elevated CER and WER rates for such text.

When we examine the dataset-specific results, Jeannette consistently outperforms Albatross, especially in handwritten text. This can be attributed to two primary factors: firstly, Jeannette features a more uniform writing style, whereas Albatross exhibits a higher stylistic variability. Secondly, Albatross’s tables generally contain more text-filled cells, leading to increased text density and heightened challenges in text detection and recognition.

6.3.2 Structure Detection

We employ GNNs outlined in Section 6.1 to identify substructures like rows and columns. These GNNs classify the edges of an initial graph, and a connected components algorithm is then applied to extract the desired substructures.

Although each is trained independently, the model’s architecture remains consistent for both row and column detection tasks. Specifically, the model comprises four *EdgeConv* [Wan+19] layers, each featuring a 64-neuron MLP. A separate MLP is used for the final classification, using the features generated by the *EdgeConv* layers as input. This classification MLP consists of four layers, each with 64 neurons, and a terminal binary layer that employs a *Sigmoid* activation function. Each model is trained for 4 000 iterations, utilizing *Cross Entropy* as the loss function.

6. Information Extraction in Structured Documents

Table 6.2: F_1 for structure recognition. J+A refers to Jeannette and Albatross, both at the same time.

Corpus	Jeannette		Albatross		J+A	
Test type	GT	Hyp	GT	Hyp	GT	Hyp
Rows	98.16	98.22	98.21	85.77	93.03	88.66
Cols	97.94	95.61	97.74	87.72	93.08	89.41

In line with the techniques presented in [PV21], we have set specific parameters for the initial graphs used in row and column detection. For row detection, the parameters are $\sigma_1 = 4$, $\sigma_2 = 0$, $s_h = 1$, and $s_w = 1$. For column detection, they are $\sigma_1 = 0$, $\sigma_2 = 4$, $s_h = 1$, and $s_w = 1$.

Figure 6.7 illustrates the pipeline, starting from an initial graph. Utilizing a GNN in conjunction with an MLP, followed by a connected components algorithm, we can extract various substructures.

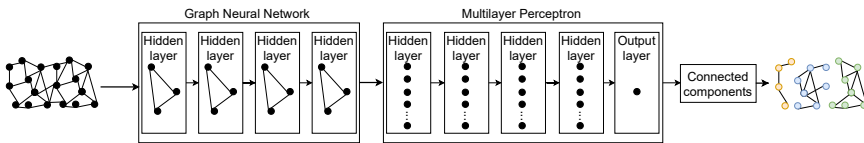
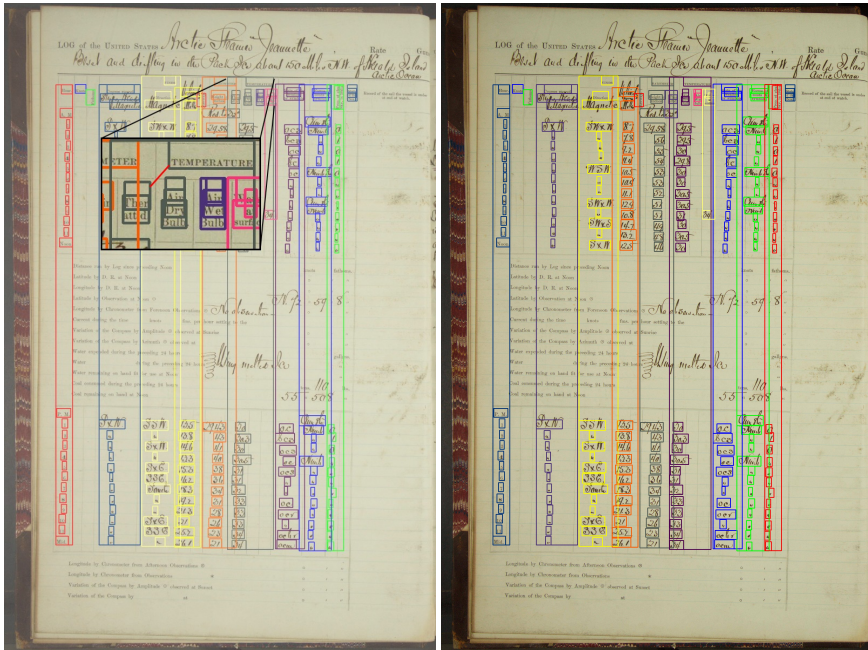


Figure 6.7: Visualization of the pipeline, using a GNN, for the structure extraction.

Table 6.2 presents the performance metrics for row and column detection using both Jeannette and Albatross datasets and a combined training approach. These results, measured with the metric explained for this purpose in Section 6.2, the structure F_1 , measure how well the structures have been completely extracted, meaning without missing any of the textlines, either by including one that should not be there or leaving one out, regardless of their size.

Jeannette’s results are expectedly high, achieving an structure F_1 score of 98.16 for rows and 97.94 for columns. This is expected, given Jeannette’s relatively simple and consistent table structures. The GNN model successfully identifies nearly all structures in this dataset. Similar high performance is observed in Albatross when using reference lines, despite its less homogeneous table structures and at least seven different table types with minor variations (such as the swap of a column, for example).

When training on both Jeannette and Albatross datasets simultaneously (J+A), the structure F_1 score slightly declines to around 93 for both rows and columns. This drop is attributed to the increased variability in table structures. Nonetheless, the performance remains commendably high, indicating the robustness of our approach.



(a) Detected textlines.

(b) Reference textlines.

Figure 6.8: Automatically detected structures at the left and reference textlines at the right. We can see the missclassified edge, zoomed, at the left. This edge joins two different columns.

In the “Hyp” columns of Table 6.2, we observe the performance metrics when using hypothesis lines generated by MaskRCNN. Compared to the reference pages, these lines are labeled based on similarity metrics, primarily IoU. Although this approach provides a quasi-GT, it is worth noting that these metrics serve as approximations rather than definitive measures. For a more accurate evaluation, manual labeling by experts would be required, which is both time-consuming and expensive.

In Jeannette’s case, the performance with hypothesis lines remains

6. Information Extraction in Structured Documents

largely consistent with that of the reference lines, showing a minor decline of just over 2 of structure F_1 points for columns. Conversely, Albatross experiences a more significant drop, losing nearly 13 and 10 structure F_1 points for rows and columns, respectively. This decline is likely due to the complexity of automatically detecting lines in Albatross, compounded by the higher density of handwritten text in its tables. When both corpora are combined, the performance metrics slightly improve over Albatross alone, reaching up to 88 and 89 structure F_1 points for rows and columns, respectively. This suggests that including Jeannette’s corpus has enhanced the training efficacy. This combined approach is particularly noteworthy as it challenges the GNNs to generalize across different datasets and provides a more extensive test set for evaluation. Achieving structure F_1 scores close to 90 is considera

It is important to clarify that a high structure F_1 score for structure extraction does not necessarily guarantee effective information extraction. For example, even if the structure F_1 score is high but still does not reach 100 points, it can still extract all relevant information accurately. Some errors, such as merging parts of column headers, can be rectified in later stages, perhaps by a language model that can still identify the relevant query for the header. Similarly, merging headers of empty columns may reduce the metric but will not result in information loss.

However, there are severe, irreparable errors, like losing a column header and failing to link it with its corresponding column. Such errors lead to the loss of crucial information that cannot be recovered in subsequent stages. This structure-detection approach is particularly sensitive to false positives (FPs), which are highly undesirable. An FP essentially classifies as correct an edge that should not exist, causing two substructures to merge and leading to information loss and contamination. In Figure 6.8, it is depicted as an example of an FP joining two columns that should not be joined. On the left, we see, zoomed in, the part where the error is occurring, merging the “Temperature” column with the “Ther attd” column, marking in red the wrongly classified edge in this case. On the right, we can see the reference column groupings.

Table 6.3: Classification Error Rate of header classification. J+A refers to Jeannette and Albatross, both at the same time.

Corpus	Jennette		Albatross		J+A	
Test type	GT	Hyp	GT	Hyp	GT	Hyp
% Classif. Error	0.64	1.94	0.65	0.62	0.48	1.09

6.3.3 Header Detection

For the header detection task, we use GNNs to classify the nodes in a binary manner, as we have explained in Section 6.1. The detection of row headers is not done by classifying nodes but is left for the end of the IE process, relying on some heuristics, as these headers are always times of the day and are always in the first column. We need something more sophisticated for the column nodes classification, as although they are usually printed, sometimes there are strikethroughs, and they are manually rewritten. Additionally, the layout of the header itself, marked by the tables, is not usually respected.

For this task, we have used a GNN composed of 5 *EdgeConv* layers [Wan+19], which expect an input MLP to process the data for each node. Each MLP consists of a single layer of 64 neurons, each with *Mish* [Mis19] as the activation function, with the last one having an output for a single neuron with Sigmoid as the activation function. The model has been trained for 2 000 iterations using *Cross Entropy* as the loss function. In this case, following the techniques presented in [PV21] for the creation of the initial graph, the parameters $\sigma_1 = 0$, $\sigma_2 = 0$, $s_h = 1$, and $s_w = 1$ have been set, where basically the parameters are being disabled and a graph is being created using only the line-of-sight of each node, given that the problem to be solved is more straightforward and does not require as many edges.

As indicated in Table 6.3, the classification error is generally below 1%, making this task relatively straightforward. Both Jeannette and Albatross datasets yield similar and notably good results when using GT lines. There is a slight but statistically insignificant improvement with the hypothesis lines in Albatross. In contrast, its performance in Jeannette deteriorates by nearly two percentage points. When both corpora are used simultaneously for training, the results improve, suggesting that increased data availability

6. Information Extraction in Structured Documents

Table 6.4: Classification Error Rate of span edges. J+A refers to Jeannette and Albatross, both at the same time.

Corpus Test type	Jenanette		Albatross		J+A	
	GT	Hyp	GT	Hyp	GT	Hyp
% Classif. Error	0.09	4.89	0.07	5.41	0.09	5.09

enhances the model’s performance.

It is worth noting that most errors in header classification are not necessarily final. Even with partial header information, the language model employed in the final stage of the IE process can potentially correct these errors, allowing for successful information extraction.

6.3.4 Span Detection

Once we have identified the headers of each table, we can proceed to detect their spans. It is important to note that in this specific corpus –Jeannette and Albatross – spans are only present in the headers. Therefore, we simplify the initial graph by focusing exclusively on the detected headers. However, this same approach can be applied to other corpora without this specific condition.

Detecting spans in the headers enables us to execute more complex and specific queries, freeing us from the constraint of relying solely on words that appear in individual cells. This allows us to base our queries on words from multiple cells that are “linked” as spans by the table’s inherent structure.

For this task, the initial graph was constructed using only the detected headers, resulting in a much smaller graph than usual and consequently speeding up the model training process. The model architecture remains the same as described in Section 6.3.2, utilizing four-layer *EdgeConvs* with 64-neuron MLPs and a final MLP for edge classification. The key difference between this graph and the one in Section 6.3.2 is that the graph here is directed, as outlined in Section 6.1. Following the techniques in [PV21], the parameters for the initial graph were set to $\sigma_1 = 1$, $\sigma_2 = 3$, $s_h = 3$, and $s_w = 3$, based on the training data and the reduced complexity of the graph when focusing exclusively on headers.

In Table 6.4, we can observe the classification errors for each task. When using the GT corpora, the error rate is almost negligible, registering less than 1% in classification errors. However, when relying on automatically detected and labeled lines, the error rate increases to approximately 5%. This could be attributed to inaccurately detected lines, leading to incorrect labeling during the creation or estimation of the Ground Truth.

6.3.5 Information Extraction

We have seen how, step by step, we have solved the subproblems presented to be able to extract information.

It is worth mentioning a special case that often occurs not only in these corpora but also in many other corpora of tables or forms, and that is the case of quotation marks. Sometimes, we may find that the writer, instead of writing the content of a cell, wrote the content of a quotation mark, usually referring to the content being the same as in some previous cell. To address this, we have employed row and column graphs. Once these graphs identify the rows and columns, they automatically define the cells at their intersections. If a cell contains a quotation mark, we can straightforwardly copy the content from the preceding cell in the same row, resolving the issue.

Now, let us consolidate all the results and report the information extraction from the content of the tables using IE F_1 .

Table 6.5 provides a comprehensive view of our IE performance under different conditions: a) with both lines and transcriptions being GT; b) with GT lines but HTR-generated transcriptions; and c) with both lines and transcriptions generated automatically.

This evaluation is closely related to the WER we discussed earlier. However, its impact varies because not all information on a page is tabular, and the system must distinguish between tabular and non-tabular content. The most challenging scenario involves using both automatically detected lines and HTR-generated transcriptions. This approach is the most reflective of real-world conditions but is also most susceptible to cumulative errors throughout the process.

We provide a detailed breakdown of our results for the Jeannette, Albatross, and combined (J+A) corpora in each test scenario. To measure how closely our method approximates ideal IE, we include a theoretical “Oracle”

6. Information Extraction in Structured Documents

performance in Table 6.5. This Oracle row reveals that no errors would be introduced when using GT for both lines and content. We also present the highest achievable performance when utilizing transcriptions generated by the HTR system.

It is important to note that the correlation with the WER, as shown in Table 6.1, is not straightforward. This discrepancy is mainly due to the treatment of quotation marks. While their accurate transcription does not influence the WER, it can adversely affect IE performance if the referenced text is inaccurately transcribed.

Lastly, when combining line detection with HTR, the best achievable performance, given by the Oracle, yields IE F_1 scores of 0.86 for Jeannette, 0.66 for Albatross, and 0.73 for the combined (J+A) corpora.

We observe that GNNs, when utilizing GT content, achieve an IE F_1 score exceeding 0.90, with Jeannette yielding the best results. However, when hypothesis-based content is introduced, the performance drops. Specifically, the reduction ranges from 0.04 in IE F_1 in the best-case scenarios involving Jeannette and the combined J+A corpus, to 0.07 in IE F_1 in the worst-case scenario with Albatross. In a more realistic setting, where both lines and content are generated as hypotheses, the performance drops by 0.07 in IE F_1 for Jeannette and improves slightly to a 0.03 drop in IE F_1 for Albatross. Interestingly, when using the combined J+A corpus for both training and testing, the performance tends to hover just above the unweighted average of the separate Jeannette and Albatross results. This suggests that the model benefits from a larger training dataset and exhibits better generalization capabilities.

Furthermore, Table 6.5 explains how each stage of the complete processing pipeline incrementally impacts performance. Jeannette’s drop is relatively modest, with a drop of 0.07 in IE F_1 due to HTR content errors and a similar decline when using autonomously detected lines. In contrast, the Albatross dataset, characterized by its inherent variability in layout and handwriting styles, experiences a more significant performance drop. Specifically, there is a reduction of 0.25 in IE F_1 when applying the HTR system and an additional reduction of 0.09 in the “Oracle” category when using autonomously identified lines.

These results indicate the critical importance of both line detection and HTR in the overall pipeline. Inaccurate line detection or poor transcription

Table 6.5: Information extraction F_1 results. 95% confidence intervals are never larger than 0.01. J+A accounts for Jeannette plus Albatross.

Corpus	Jeannette			Albatross			J+A		
	GT	GT	Hyp	GT	GT	Hyp	GT	GT	Hyp
Content	GT	Hyp	Hyp	GT	Hyp	Hyp	GT	Hyp	Hyp
GNN	0.95	0.88	0.81	0.90	0.68	0.65	0.93	0.76	0.70
Oracle	1.00	0.92	0.86	1.00	0.75	0.66	1.00	0.80	0.73

can introduce errors that propagate through subsequent stages, ultimately compromising the pipeline’s ability to provide accurate results.

6.4 Discussion

One of the most notable advantages of employing GNNs for table information extraction is their adaptability. With minimal model adjustments, GNNs can be tailored to handle various types of tables, forms, or similar challenges. This flexibility sets them apart from other methods requiring substantial modifications to accommodate different tabular layouts [Pri+23a]. While we have demonstrated their effectiveness using two specific datasets, these techniques can be easily generalized to other data corpora with minimal changes.

However, this versatile approach has its drawbacks. For instance, a single misclassification of an edge in a structure can result in the failure to accurately detect two separate structures, as illustrated in Figure 6.8. The current model employs a threshold for edge inclusion, which could be enhanced through more sophisticated techniques. This could involve considering the likelihood of an entire detected structure, an area that deserves further investigation.

Additionally, the focus has predominantly been on geometric attributes in the existing methods. Incorporating textual properties could significantly boost performance, although this would necessitate additional research. However, such integration would also make the GNNs in the pipeline, as seen in Figure 6.5, reliant on HTR.

At the same time, GNNs offer a level of flexibility and extensibility that is particularly advantageous. Utilizing an undirected graph, they can identify

6. Information Extraction in Structured Documents

structures like rows and columns, and they can also classify nodes into various elements, such as headers, depending on the specific requirements of the task.

Our preliminary findings indicate that GNNs can achieve impressive performance levels in IE tasks. This is especially noteworthy when compared to Oracle benchmarks and various hypotheses. However, there is room for improvement, mainly when dealing with more challenging datasets. One significant bottleneck to optimal performance is the accurate detection and transcription of lines, which can lead to a loss of up to 0.25 points in IE F_1 metrics, as observed in the Albatross dataset.

One potential avenue for research could be the development of an end-to-end system that seamlessly integrates line detection, transcription, and information extraction. Such an approach could offer computational and time efficiencies. However, the feasibility of training such a comprehensive system with limited resources – such as the Nvidia GTX 2080 with 8Gb of VRAM used in our tests – remains a concern.

In summary, while GNNs present a promising methodology for information extraction from tabular documents, there are still numerous challenges and opportunities that warrant further investigation.

7

Conclusions

In this concluding chapter, we present overall conclusions about our findings, as well as an overview of the publications and projects to which we have contributed. We also provide a list of repositories where we have made the code publicly available. Additionally, we outline the avenues for future research that this thesis has opened up.

Our research journey began with an in-depth analysis of various unresolved challenges in the field of historical documents. We identified specific areas that lacked effective solutions and had never been approached in the manner we have undertaken.

Firstly, we analyzed that the task of segmenting entire historical books – a task that requires a contextual understanding that goes beyond individual pages – had never been previously proposed. Secondly, we introduced a novel solution for classifying large collections of untranscribed historical images, another problem that had not been addressed before. Lastly, we have addressed the problem of information extraction in historical meteorological tables with a lightweight yet robust method.

In historical book segmentation, we have introduced a first-of-its-kind formalization of the problem. Our approach suggests that to achieve an optimal solution, operating at multiple levels is essential. Initially, we focused on individual image analysis. Then, using these "raw" results without making any immediate decisions, we incorporated the broader context of the entire book. This allowed us to segment the book into notarial acts or other relevant units, depending on the specific corpus under study. To tackle this complex issue, we employed a range of deep learning models, including CNNs, RPNs, and transformers, among others.

Our segmentation approach has been rigorously tested across two distinct sub-problems within the broader challenge of historical book segmentation. The first sub-problem involves segmentation with minimum

7. Conclusions

size constraints for each notarial act, requiring them to span at least two pages. This constraint simplifies the problem, making it easier to develop a solution. Through exhaustive testing, we have demonstrated that our segmentation results are both good and consistent under these conditions. For the second sub-problem, we introduced three datasets without such size constraints, where the task also includes transcribing the text on each page, not just segmenting the book. We evaluated two different HTR techniques for transcribing specific regions, which have emerged as the most effective solution for historical book segmentation as of now. Lastly, we compared this technique against the methods used in the first, size-constrained dataset on a final dataset. Our results indicate that regardless of dataset restrictions, both approaches can yield satisfactory outcomes.

Following the segmentation work, we addressed the challenge of classifying non-transcribed historical handwritten documents. These documents are part of one of the corpora we segmented earlier, specifically two books of deeds from the AHPC collection. Given that these books are not transcribed, we developed a new probabilistic framework that leverages PrIx to classify documents while accounting for the uncertainty inherent in the HTR model used for transcription. We compared this approach with traditional classification methods that rely on transcription results, the standard practice for non-handwritten text. Additionally, we employed the OSC framework and compared various techniques within it. One of the unique challenges with historical documents is that the complete set of possible classes is often unknown, leading to the emergence of new classes over time. Using OSC, we could accurately identify documents that do not belong to any predefined classes, demonstrating the framework's effectiveness.

Lastly, we tackled the challenge of information extraction in structured handwritten documents, such as tables or forms. We developed a comprehensive pipeline that starts with the raw image of the document. The first step in our pipeline involves text detection, for which we employed RPNs to identify lines of text. We transcribe these lines using a line-level model and identify various substructures like rows and columns to reconstruct the complete table structure. We also addressed challenges, such as handling multi-span columns and quote marks. Our solution enables the detection of table headers and facilitates the execution of queries, allowing for end-to-end evaluation at different pipeline stages. For this task, we utilized

RPNs for text detection, CRNNs for text transcription, and GNNs for detecting and reconstructing the complete table structure. This multi-faceted approach ensures robust and accurate information extraction from complex handwritten tabular documents.

7.1 Scientific Outcomes

The outcomes of this thesis have led to multiple scientific publications, including 9 conference papers and three journal articles, with an additional one pending publication. The following section categorizes these publications according to their respective research areas.

- **Book and act segmentation:** in the first paper, we analyze for the first time the use of textual features along with visual ones for the segmentation of acts on a single page. In the following two papers, we segment the entire book to extract notarial acts from handwritten books, going beyond the physical page. Additionally, we provide a new metric to measure these results.
 - **Prieto, J.R.**, V. Bosch, E. Vidal, D. Stutzmann and S. Hamel, “Text Content Based Layout Analysis”, In: 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 2020, Pages 258-263
 - **Prieto, J.R.**, Becerra, D., Toselli, A.H., Alonso, C., Vidal, E. (2023). “Segmentation of Large Historical Manuscript Bundles into Multi-page Deeds”. In: Pertusa, A., Gallego, A.J., Sánchez, J.A., Domingues, I. (eds) In: Pattern Recognition and Image Analysis. IbPRIA 2023. Lecture Notes in Computer Science, vol 14062. Springer, Cham.
 - **Prieto, J.R.**, Becerra, D., Toselli, A.H., Alonso C., Vidal E., “Segmenting Large Historical Notarial Manuscripts Into Multi-page Deeds”. In: Pattern Analysis and Applications vol 27, n° 22 (2024)

7. Conclusions

- Non-transcribed Historical Document Classification: in this series of papers, we focus on developing a framework specifically designed to classify non-transcribed historical documents. Initially, we applied this technology to a project known as Carabela. As we progressed, we continued to refine and enhance the technology, including integrating the OSC framework to improve the classification process further.
 - **Prieto, J.R.**, Flores, J.J., Vidal, E., Toselli, A.H, “Open set classification of untranscribed handwritten text image documents”, In: Pattern Recognition Letters, Volume 172, 2023, Pages 113-120
 - Flores, J.J., **Prieto, J.R.**, Garrido, D., Alonso, C., Vidal, E. (2022). “Classification of Untranscribed Handwritten Notarial Documents by Textual Contents”. In: Pinho, A.J., Georgieva, P., Teixeira, L.F., Sánchez, J.A. (eds) Pattern Recognition and Image Analysis. IbPRIA 2022. Lecture Notes in Computer Science, vol 13256. Springer, Cham
 - E. Vidal; Romero, V.; Toselli, A. H., Sánchez, J.A., Bosch, V., Quirós, L., Benedí, J.M, **Prieto, J.R.**, “The Carabela Project and Manuscript Collection: Large-Scale Probabilistic Indexing and Content-based Classification”, 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 2020, Pages 85-90
- Extraction of Descriptive Words from Non-Transcribed Documents: While this aspect has not been covered in the current document, the advancement of classification techniques has also led to research on methods for extracting descriptive words from books. This auxiliary research complements the primary focus on document classification.
 - **Prieto, J.R.**, Vidal, E., Sánchez, J.A., Alonso, C., Garrido, D. (2022). “Extracting Descriptive Words from Untranscribed Handwritten Images”. In: Pinho, A.J., Georgieva, P., Teixeira,

L.F., Sánchez, J.A. (eds) Pattern Recognition and Image Analysis. IbPRIA 2022. Lecture Notes in Computer Science, vol 13256. Springer, Cham.

- Classification of Writing in Documents: Although this has not been explored in this document, the research line of author identification using CNNs has been explored.
 - Punjabi, A., **Prieto, J.R.** and Vidal, E., “Writer Identification Using Deep Neural Networks: Impact of Patch Size and Number of Patches”, In: 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021, Pages 9764-9771.
- Information Extraction in Historical Tabular Datasets: In subsequent papers, we focused on extracting information from tabular data found in handwritten historical images. Initially, we enhanced the existing technology for identifying substructures within graphs, laying the groundwork for future research. Following this, we applied our improved methods to the HisClima project. Here, we made several contributions that further refined the GNN-based technology for information extraction.
 - **Prieto, J.R.**, Vidal, E. (2021). “Improved Graph Methods for Table Layout Understanding”. In: Lladós, J., Lopresti, D., Uchida, S. (eds) Document Analysis and Recognition – ICDAR 2021. ICDAR 2021. Lecture Notes in Computer Science(), vol 12822. Springer, Cham.
 - **Prieto, J.R.**, José Andrés, Emilio Granell, Joan Andreu Sánchez, Enrique Vidal, “Information extraction in handwritten historical logbooks”, In: Pattern Recognition Letters, Volume 172, 2023, Pages 128-136
 - Andrés, J., **Prieto, J.R.**, Granell, E., Romero, V., Sánchez, J.A., Vidal, E. (2022). “Information Extraction from Handwritten Tables in Historical Documents”. In: Uchida, S., Barney, E.,

7. Conclusions

Eglin, V. (eds) Document Analysis Systems. DAS 2022. Lecture Notes in Computer Science, vol 13237. Springer, Cham.

- Granell, E., Romero, V., **Prieto, J.R.**, Andrés, J., Quirós, L., Sánchez, J.A. and Vidal E., “Processing a large collection of historical tabular images”, In: Pattern Recognition Letters Volume 170, 2023, Pages 9-16

7.2 Projects

Some of the methods and models developed during this thesis have been used in projects where a large number of images have been processed.

- *Simancas Search*: This is an ongoing project where we have focused on a range of tasks. These include detecting lines and layouts within historical documents, as well as segmenting notarial acts. The segmentation approach we used is detailed in Chapter 4, and the results are presented in Section 4.5.
- *Carabela*: This project was the first to apply the probabilistic framework for the classification of non-manuscript historical images.
- *HisClima*: In this project, beyond processing the layout of the images, work was done on information extraction from tabular content. Assistance was also provided in the creation of the dataset.

7.3 Open Source Software

In both the academic and scientific communities, especially within Computer Science, open-source software has emerged as a crucial enabler for progressing our collective knowledge. This is particularly true in rapidly evolving and intricate fields like artificial intelligence. The act of making code publicly available serves multiple purposes: it increases the transparency of the research, allows for its reproducibility, and encourages collaborative efforts that can lead to accelerated innovation.

When it comes to the specialized application of AI techniques to handwritten historical documents, the role of open-source software becomes

even more significant. It offers a dependable and easily accessible framework, enabling researchers across various disciplines to validate and build upon existing work efficiently.

Given these compelling reasons, we have made the code used in this thesis publicly available.

- **Classification of Non-transcribed Documents.** The code associated with the experiments detailed in Chapter 5 has been made publicly accessible.

<https://github.com/JoseRPrietoF/docClassPrIx>

- **Information Extraction from Historical Tables.** The source code employed for creating graphs, detecting table structures, and extracting information – detailed in Chapter 6 – is available to the public.

<https://github.com/JoseRPrietoF/tableIE>

7.4 Future Work

In this section, we explore potential avenues for future research in each area investigated in this thesis. Like many other scientific domains, there are ample further study and development opportunities.

7.4.1 Whole Book Segmentation

We believe that this area of research presents the most opportunities for further exploration. This thesis marks the first time that the complete segmentation of historical handwritten books has been proposed, offering various approaches based on the specific constraints of the studied books.

However, the decoding methods outlined in Section 4.1 could be enhanced by incorporating more advanced language models using the SRILM toolkit [Sto02]. Moreover, the most promising results achieved using RPN do not currently leverage textual content for region identification and subsequent decoding. While these models may be capable of visually discerning

7. Conclusions

words, incorporating textual features alongside the image could improve RPN performance.

We are optimistic about the potential of transformer-based models, especially given their recent successes in various fields [Isl+23]. These models are increasingly capable of working with smaller datasets and leveraging pre-trained models. Although this has been challenging with historical data due to its divergence from modern datasets, recent advancements may change this landscape [PP23]. Using a large attention window for book segmentation could allow the model to consider text from multiple pages, thereby improving segmentation accuracy. One approach could be to employ the caching technique used in [CCP23] to retain text from previous steps. This, combined with recent advancements in expanding the attention window, could enable a much larger feature window for textual characteristics [PSL22].

7.4.2 Open Set Document Classification

In this research area, we identify incremental learning as the first avenue for further exploration [GHC21; LPR22; LK22]. This approach would naturally extend the OSC framework, allowing us to identify classes outside of a predefined set and expand this set with newly detected and expert-labeled classes.

Another promising direction involves modeling the internal structure of documents to handle increasingly complex image-based documents. Future work could explore alternative classification models, such as RNNs, to capture the sequential patterns found in the textual content of consecutive document pages.

Lastly, we see potential in combining book segmentation with classification beyond the scope of classifying non-handwritten historical documents. Features useful for classifying texts could likely improve the segmentation of the book, thereby integrating these two research tasks into a unified approach.

7.4.3 Information Extraction from Historical Tabular Data

In Information Extraction from handwritten tables, we see an untapped opportunity in leveraging textual content for substructure detection using

GNNs. Recognizing that text in such documents is often challenging to transcribe, we propose using PrIx as a solution (see Section 3.2.1). PrIx offers rich probabilistic word distributions and their geometrical position, unlike traditional HTR methods that yield single, error-prone transcriptions. This approach could help avoid making irreversible decisions during the data processing stages.

Taking it a step further, an even more ambitious approach would be to develop an end-to-end system. This unified model would handle detection, transcription, and structure extraction, concurrently utilizing both textual and visual content. While similar efforts have been made in other research areas [Li+22a], none have tackled the unique challenges posed by handwritten text and historical tables with diverse layouts.

7.4.4 The Era of Large Language Models

We are currently experiencing a surge in the field of Large Language Models (LLMs), which are models with millions of parameters extensively trained on vast datasets, usually by major companies. Starting with BERT [Dev+19], the scientific community saw significant improvements in NLP results. Researchers started using this pre-trained model and fine-tuning it for better outcomes. Subsequent transformer-based models like T5 [Raf+20] pushed the boundaries of transfer learning even further. LLMs continued to grow in size, leading to models with billions of parameters and increasingly massive datasets for training, extending beyond text to include visual content, as in the case of CLIP [Rad+21].

However, as researchers increasingly focused on larger models and indiscriminately extracted vast amounts of data from the internet, Hoffmann et al. [Hof+22] demonstrated that data quality is equally or more important than model size.

In the wake of this shift, the community began training models and releasing them as open-source. Models ranging from just over 1 billion to 70 billion parameters have been released [Tou+23; Jia+23] and democratized access to pre-trained LLMs, albeit with computational constraints. Thanks to these models and platforms like HuggingFace [Wol+20a], the community has been able to utilize pre-trained LLMs.

However, these models remain prohibitive for most researchers due to the computational resources required for training and inference. Re-training

7. Conclusions

or fine-tuning these models is often slow and tedious. Some advances have been made in reducing memory and computational costs by reducing precision from 32 or 16 bits to just 8 [Det+22] or even 4 bits [DZ23]. While this usually worsens the results, the trade-off may be worth it, especially when resource limitations prevent model execution.

Given this context, we believe that a potential research avenue is opening up (or at least worth exploring) using pre-trained open-source LLMs to adapt them to the problems presented in this thesis. For text classification problems, positional encodings could be adapted for use with PrIx, which transformer-based models use to indicate the position of each word. For book segmentation, as mentioned earlier in Section 7.4.1, using LLMs could provide a larger attention window across the entire document and richer feature extraction due to the fusion of visual and textual information. Finally, for information extraction, models like Donut [Kim+22] or Nougat [Ble+23] perform IE on non-handwritten documents. Adapting these models to historical documents could be viable thanks to recent advances and the increasing availability of handwritten table datasets.

Appendices

A

Datasets

Throughout the development of this thesis, various datasets have been used and processed. Most of them are publicly available to facilitate the replication of the results presented. In this section, we provide a link to access these datasets when they are available, along with a description.

A.1 Alcar - HOME

Three folders from the HOME-Alcar project have been utilized: Nesle, Denis, and Navarre. Each of these folders contains a series of notarial acts distributed among their images and has a distinct layout from the others. These notarial acts contain information dictated by the reigning king at the time, and their length can vary from a paragraph to several pages. These datasets are particularly interesting for the task at hand, which is to sequentially segment this information throughout a book, preparing the data for future information extraction.

The train, validation, and test splits in each corpus have been carefully hand-selected to avoid cutting any acts, simulating a complete book with its beginning and end.

A.1.1 Nesle

The Nesle seigneurie cartulary (Côte-d’Or, cant. Laigne) comprises a compact book comprising 117 well-preserved parchment folios. Spanning the years 1217 to 1282, it features 81 notarial acts in Latin and Middle French. Algunas imágenes en blanco o sin actos notariales consecutivas des del principio del libro se han descartado. As displayed in Table A.1, 96 images have been selected, distributed as 68 for training, 8 for validation, and 20 for testing. The table also presents the allocation of documents

A. Datasets

Table A.1: Number of pages and acts in the Nesle, Denis, and Navarre folders.

	Nesle				Denis				Navarre			
	Train	Val	Test	Tot.	Train	Val	Test	Tot.	Train	Val	Test	Tot.
Img.	68	8	20	96	129	20	50	199	124	27	53	205
I	56	6	13	75	196	24	66	286	58	5	29	92
M	62	8	22	92	40	14	30	84	56	21	19	96
F	56	6	13	75	196	24	66	286	58	5	29	92
C	4	0	2	6	98	7	17	122	21	0	11	32
Acts	60	6	15	81	294	31	83	408	79	5	40	124

across these different partitions, as well as their categorization under the “IMFC” sequence.

Each manuscript image displays a pair of pages, with a typical single column extending across each page, as illustrated in Figure A.1. This figure presents images identical to those in Figure 3.5, albeit without markings. The documents within the cartulary are distinctly demarcated by Roman numerals, with each act commencing with a capital letter. However, these Roman numerals and capital letters are not annotated in the Ground Truth (GT) and, therefore, despite their presence in the image, are excluded from the transcribed text. The images contained in this collection consistently adhere to the aforementioned layout.

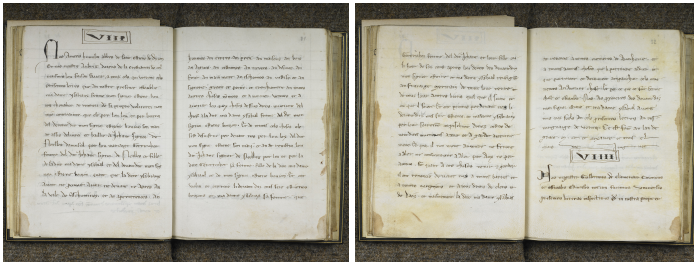


Figure A.1: Example of two consecutive pages from the Nesle corpus.

A.1.2 Denis

The Denis documents comprise a set of two volumes produced between the late 1240s and the 1300s. Each image consists of two columns, as

exemplified in Figure A.2, which displays two consecutive pages labeled with the "IMFC" sequence. The dataset is partitioned into 129 images for training, 20 for validation, and 50 for testing, as shown in Table A.1.

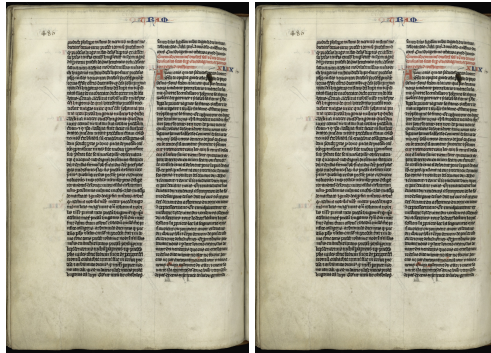


Figure A.2: Example of two consecutive pages from the Denis corpus.

In this case, the typography is more intricate than in the previous book. Each notarial act begins with a few lines in a different color, usually red, and is typically accompanied by a capital letter. Occasionally, letters or figures can be found between acts or within the same act, as seen in Figure A.2, on the final line of the first column. These elements are not annotated in the GT since only the text and the beginning and end of each act are recorded.

A.1.3 Navarre

The Navarre Cartulary bears witness to the history of Charles II, Count of Évreux (1343-1378) and King of Navarre (1349-1387). Each image features a single page with one column per page, as illustrated in Figure A.3. The typography in this collection is even more challenging than in the two previous cases, making it visually harder to distinguish individual notarial acts. Typically, a left margin is left blank for the opening lines of a notarial act, although this is not always the case. The dataset is partitioned into 129 images for training, 20 for validation, and 50 for testing, as shown in Table A.1.

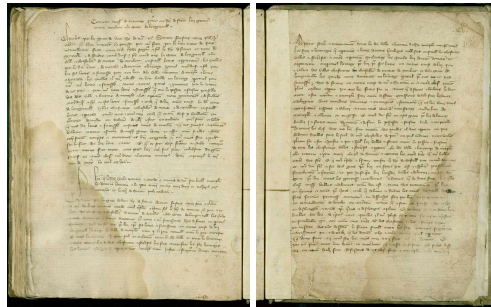


Figure A.3: Example of two consecutive pages from the Navarre corpus.

A.2 Archivo Histórico Provincial de Cádiz (AHPC)

The JMDB Series of Notarial Record Manuscripts is housed in the Spanish Provincial Historical Archive of Cádiz (AHPC), established in 1931 to collect and safeguard notarial documents that are over a century old. The AHPC's functions include preserving the provincial documentary heritage and providing researchers access to these invaluable resources. This dataset belongs to the CARABELA collection [Vid+20]. The immense variety and ever-changing writing styles, extensive use of archaisms and non-standard abbreviations [Rom+19b], the subpar quality of original documents and/or scanned images, and the sheer volume of the collection make CARABELA one of the most challenging sets of historical manuscripts we have ever encountered.

The dataset under consideration in this subchapter is derived from an extensive collection of 16,849 "notarial protocol books," each containing approximately 250 notarial deeds or acts and averaging 800 pages. Additionally, each notarial act is assigned a specific class.

The JMDB Series comprises notarial records produced by Juan Manuel Briones Delgado, a notary in Cádiz between 1712 and 1726. Each AHPC bundle is systematically arranged into consecutive sections, each corresponding to a deed or notarial act, except for an initial section consisting of roughly 50 pages that serve as a table of contents for the bundle. This introductory section was identified but not utilized in the experiments.

In this series, the deeds are "page-aligned," with each deed commencing on a new recto page and spanning anywhere from one to several dozen

pages, some of which may be (almost) blank.

The first and last pages of each deed can often be easily distinguished visually due to minor layout differences compared to other pages. However, separating the deeds within a bundle is a complex task, as many regular pages may bear striking similarities to initial and/or final pages, leading to confusion. As described in the previous chapter, Figure 3.4 displays a typical deed comprising initial, final, and four regular mid-page images.

With this dataset and the corresponding annotated GT, we can undertake two tasks: act segmentation, with four available folders containing annotated GT, and act classification, with two folders.

A.2.1 Act Segmentation

The JMDB Series of Notarial Record Manuscripts includes 50 bundles incorporated into the collection compiled by the CARABELA project [Vid+20]. From these, we selected four bundles—JMBD4946, JMBD4949, JMBD4950, and JMBD4952—dated between 1722 and 1726 for manual Ground Truth (GT) annotation.

Table A.2 displays the statistics for these bundles. As observed, each bundle contains over one thousand pages, except for JMBD4952, which has 980 pages. The number of deeds also varies, with over two hundred per bundle and nearly three hundred in JMBD4950. A notable challenge for segmentation arises from the variability in the number of pages between deeds. In JMBD4946, for instance, there is a variance of more than 14 pages per deed, with one deed extending up to 200 pages.

This significant variability in page length complicates automated segmentation, as methods relying exclusively on page count or structure prove insufficiently effective. Consequently, additional strategies are required to identify and separate individual deeds within a bundle accurately.

A.2.2 Act Classification

For act classification we chose two manuscripts from the collection, JMBD4949 and JMBD4950, dated from 1723 to 1724. It should be emphasized that traditional GT annotations, like text lines or transcripts, are not present for these manuscripts. Instead, only coarse-grained GT annotations targeting bundle segmentation and deed classification were generated.

A. Datasets

Table A.2: Number of page images and deeds for the bundles JMBD4946, JMBD4949, JMBD4950 and JMBD4952.

	JMBD4946	JMBD4949	JMBD4950	JMBD4952
N. Pages	1399	1615	1481	980
N. Deeds	248	295	260	236
Avg Pages per Deed	5.79	5.47	5.69	4.20
Min-max pages per Deed	2–200	2–122	2–62	2–38
St-dev pages per Deed	14.27	9.93	8.19	4.08

Specialists found 295 deeds in JMBD4949 and 260 in JMBD4950, amounting to 555 deeds associated with roughly 41 distinct types or classes. Nevertheless, the classification of some deeds remained ambiguous, and for a number of the identified classes, only a scarce quantity of deeds was obtainable. To guarantee the reliability of the classification results, only classes with a minimum of one deed in each book and six deeds in total were taken into consideration.

Consequently, 498 deeds from 12 classes regarded as adequately represented were retained, while the remaining ones, associated with 29 indistinct or underrepresented classes, were collectively designated as a unique “class” called REJECT (RJ).

The twelve well-documented classes encompass:

Power of Attorney (PA), Letter of Payment (LP), Debenture (DB), Lease (LE), Testament (TE), Sale (SA), Risk (RI), Census (CS), Deposit (DP), Statement (ST), Cession (CN), and Treaty of Fact (TF).

Specifics of this dataset can be located in Table A.3.

This corpus exhibits a peculiarity in classifying deeds into the mentioned classes. Typically, the first page of each deed contains a rectangular section along the left margin, which provides information about the type of deed being documented. Figure A.4 shows multiple examples of this.

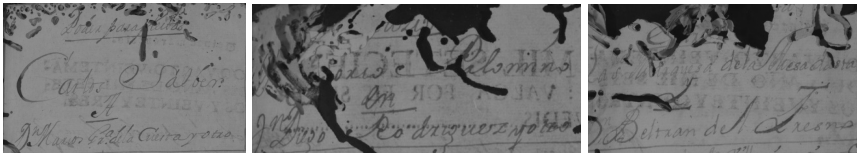


Figure A.4: Three examples of headers for the books JMBD4949 and JMBD4950. The first two are of class PA, and the last one is of class LE.

Table A.3: Number of documents and page images for JMBD4949 and JMBD4950: per class, per document & class, and totals.

Class ID's	Deeds	Pages				Total
		Avg	Min	Max	St-dev	
PA	240	3.3	2	24	3.5	803
LP	72	4.8	2	30	5.4	345
DB	44	4.8	2	32	5.6	212
LE	32	4.8	2	16	2.6	152
TE	29	8.6	4	48	9.4	248
SA	21	22.9	4	122	29.8	480
RI	17	4.0	4	4	0.0	68
CS	12	11.5	2	26	9.0	138
DP	10	3.8	2	8	1.9	38
ST	9	2.4	2	4	0.8	22
CN	6	5.3	2	14	3.9	32
TF	6	5.3	4	8	1.9	32
REJECT	57	9.2	2	70	12.2	526
Total	555	5.6	2	122	9.2	3096

Examining the statistics related to these rectangles is useful, which are generally easy to detect. While we have the GT for their layout, we do not have it for their transcription. If we considered each deed solely based on its rectangle, each would occupy just a single page (the rectangle in question) and contain very few running words, significantly reducing the overall vocabulary. In Figure A.4, we present statistics comparing the average number of RWs in these header rectangles to those in the complete deeds, broken down by class. Notably, while the headers average around 10 running words, the complete deeds contain more than 1 300 on average.

It is important to note that these header rectangles are often in deplorable condition. This deterioration is commonly due to their consistent placement at the top-left corner of a page, making them susceptible to wear and tear over time and damage from insects, among other factors. This can be observed in Figure A.4. The first rectangle, which belongs to class PA, has relatively legible text considering the corpus's condition. The second rectangle, also of class PA, is severely worn, making it difficult to distinguish words. A similar issue is evident in the third rectangle, which belongs to

A. Datasets

Class ID's	Estimated RWs	
	Complete Deed	Header Deed
PA	859.3	10.6
LP	1112.3	11.6
DB	1310.9	10.0
LE	1441.1	12.2
TE	2279.3	6.3
SA	5902.5	11.4
RI	1356.9	9.1
CS	2901.2	15.2
DP	874.5	11.2
ST	716.7	8.6
CN	1355.9	8.7
TF	1404.5	12.6
Avg. page	1345.0	10.6

class LE. Consequently, relying solely on these rectangles for classification would be challenging. Moreover, not all deeds feature these rectangles; in some cases, the text remains indistinguishable even when the paper itself is well-preserved.

A.3 Hisclima

The HisClima database is a publicly accessible handwritten dataset¹ consisting of manuscripts related to ship logbooks containing climate information from the OldWeather collection². It was compiled during the HisClima project³.

The HisClima database contains two types of pages. First, table images with handwritten daily weather conditions are recorded every hour, featuring a pre-printed template containing sea temperature, atmospheric pressure, wind direction, and more. These tables include numerous abbreviations, numeric data, quotes, and other elements distributed across cells. Second,

¹<https://zenodo.org/record/7442971>

²<https://www.oldweather.org/>

³<https://www.prhlt.upv.es/hisclima-dos-siglos-de-datos-climaticos/>

The figure displays three examples of logbook tables. The leftmost table is from the USCGC Jeannette, featuring 17 columns and a header with handwritten text. The middle table is from the USCGC Albatross, featuring 19 columns and a header with handwritten text. The rightmost table is also from the USCGC Albatross, featuring 18 columns and a header with handwritten text. Each table contains various columns for recording data, with the Albatross tables showing more complete rows than the Jeannette table.

Figure A.5: Three table examples display Jeannette on the left and Albatross in the center and right. Each image has a different number of columns: 17 in the first, 19 in the second, and 18 in the third. Furthermore, it is worth noting that all rows are filled in the Albatross pages, while in Jeannette, only one row out of three is completed. Additionally, multi-line cells are more prevalent and tend to be larger in Jeannette, which can be attributed to the fact that most table rows are left empty.

each table page image has an associated descriptive text page image describing the same day in plain text (see Figure A.5). We will not utilize these plain text pages in this thesis.

For Information Extraction (IE) from structured documents, we focus on the logbooks of two ships, Jeannette and Albatross, which sailed the Arctic Ocean between 1880 and 1920. The dataset includes page images with tabular pages and descriptive text, presenting unique layout challenges. We concentrate on tables, which contain the most valuable information for researchers. Tables are split into upper (AM) and lower (PM) sections, containing printed headers and handwritten content cells. They primarily document weather and navigation conditions, such as wind directions and atmospheric pressures. While additional forms exist between AM and PM tables, they are often left blank. A more detailed description is available in [Gra+23], and we outline certain aspects relevant to this thesis.

Some layout challenges exhibited by these documents include the use of quotation marks to avoid rewriting the contents of preceding cells in the same column, data from a cell written in vertically or horizontally adjacent

A. Datasets

cells, crossed-out column names, different numbers of rows completed in each table, vertically oriented texts, headers with handwritten contents, multi-span column headers, and varying table layouts. Examples of these challenges are shown in Figure A.6.

The HisClima dataset includes two types of GT annotations. First, each page's layout was annotated with blocks, columns, rows, and baselines. Second, a paleographer fully transcribed the text. The transcripts are labeled as *printed* or *handwritten* to enable HTR models to distinguish between the two text types. It is essential to notice that printed text is more regular and easier to learn, making it crucial to provide individual recognition results for both text types.

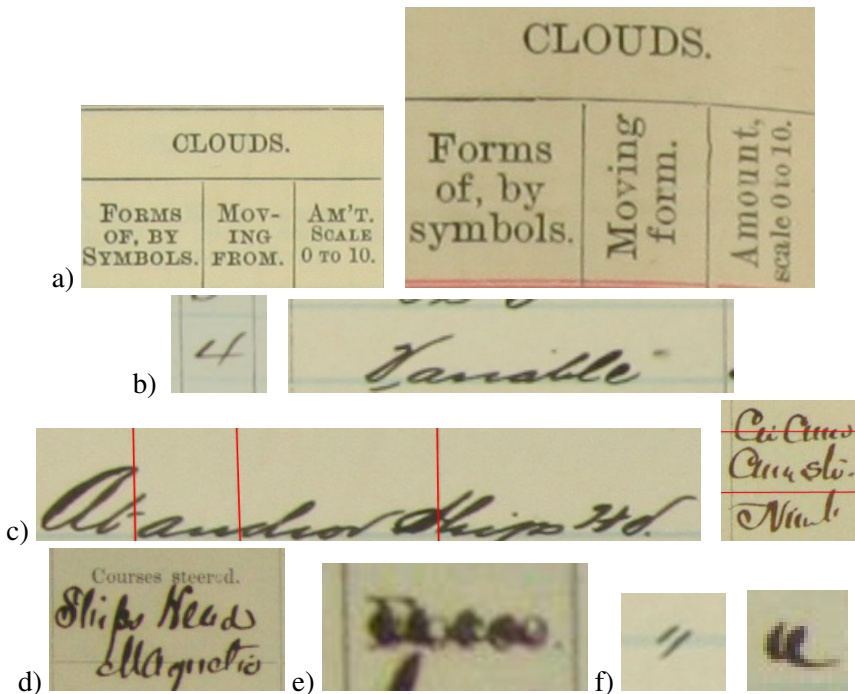


Figure A.6: Examples of challenges in the Jeannette and Albatross datasets include: a) multi-span cell headers, varying attribute writing styles depending on the table layout, and vertical text (right); b) cells with differing widths; c) cell contents that surpass boundaries (indicated in red); d) column headers containing handwritten contents; e) crossed-out column headers; and f) the use of quotation marks.

The Jeannette logbook featured a single table layout and was authored by one writer. In this logbook, climatological attributes were typically recorded every three hours, leaving several table rows empty. Examples of tables from both datasets are in Figure A.5. Additionally, multi-line cells are typical in this corpus, as can be seen in Figure A.6 c). The Albatross set comprises pages from seven distinct logbooks. In this collection, seven different table layouts have been identified. It is noteworthy that, depending on the table layout, there are variations in column headers (or slightly differently spelled versions), the number of columns, and table positions in the image. Furthermore, various writing styles are present in this set. It is important to mention that climatological attributes were typically annotated every hour in this corpus, and unlike Jeannette, most tables are fully completed. We also note that while multi-line cells are present in this corpus, they are less common than in Jeannette. The primary figures for both corpora are displayed in Table A.4. The last row, *Relevant Information (Rel. Information)*, represents the number of triplets (\hat{y}_c, h_r, v) we aim to extract (as will be explained in subsequent sections).

Table A.4: Basic statistics of the Jeannette and Albatross corpus.

Corpus Partition	Jeannette			Albatross		
	Train	Val	Test	Train	Val	Test
Pages	143	15	50	52	7	25
Lines	23 614	2 282	7 838	19 871	2 538	9 138
Rel. Information	10 923	1 014	3 561	14 123	1 764	6 420

A.4 RCSA Dataset

The Simancas General Archive, nestled in Simancas, Valladolid, stands as Spain’s premier state archive. It is the venerable official archive of the Crown of Castile, commissioned by Carlos I in 1540 within the Simancas castle. It has been the custodian of significant documents from the Crown of Castile governing councils, extending to the Hispanic monarchy and continuing to the reign of Isabel II.

Its historic trajectory mirrors the Crown of Castile’s evolution. A pivotal moment was in 1588 when Felipe II issued the Simancas Archive

A. Datasets

charter, offering insights into managing this archive and others in the region. Moreover, the archive's inflow of documents and resource allocations often echoed the highs and lows of the Castilian monarchy. The damages incurred during the War of Independence influenced the institution's modern form.

Currently, under the patronage of Spain's Ministry of Culture, UNESCO recognized its global significance, designating it a World Heritage Site in 2017.

The archive's structure encompasses diverse sections and funds, reflecting the entirety of the Crown's administrative spectrum. Notably, over half of these funds have economic underpinnings. Among its most intriguing holdings is the Books of Records of Royal Decrees collection. This collection, integral for understanding the Chamber's operation, arrived at Simancas in various consignments, initially as part of Secretary Francisco de los Cobos' 16th-century documentation. Serving as an administrative record without the Greater Seal's endorsement, these books detail royal certificates in various legal affairs, devoid of standard validation components. This includes General Books, those from Navarra, Granada, Aragon, Accountants and Finance, Military Orders, redemption of censuses, correspondence ("missives"), identity notes, and a subset from Empress Isabel de Portugal's house.

Comprising 377 books and 279,894 pages, the Books of Records of Royal Decrees collection stands as a testament to the intricate administration of the past. Figure A.7 shows some page examples of this collection.

The magnitude of this collection underscores the imperative to mine and discern vital semantic data (names, dates, locations, etc.), catering to the academic community and enriching public knowledge.

From the Royal Cedulae Simancas Archive (RCSA), as part of the Simancas Archive, experts have tagged a collection of 251 images. These images were meticulously curated into groups of consecutive images, ensuring that at least one full notarial act is present within. An act may start on one page and conclude on a subsequent page within the same group.

Of these, 190 images (representing 75% of the total) were allocated for training, while the remaining 61 images (25% of the total) were designated for testing. The distribution of acts is illustrated in Table A.5. Labels "I,M,F,C" are employed to signify the initiation, continuation, and termination of an act, particularly when an act starts on one page but doesn't



Figure A.7: Examples of some pages from the Simancas Archive.

conclude on the same. The label “C” denotes a complete act that both begins and concludes on a single page.

The “Complete” column specifies the count of entire acts within that partition. Conversely, the “Non-Complete” column denotes acts that are fragmented. Specifically, these acts aren’t tagged as “C”, neither do they commence with an “Initial Act” or “I” nor conclude with a “Final Act” or “F”, implying that we lack comprehensive information about these acts.

Table A.5: The number of acts in the Simancas Archive and their distribution.

	Complete	Non-Complete	I	M	F	C
Train	151	140	122	36	112	93
Test	52	47	42	14	34	34
Total	203	187	164	50	146	127

Table A.6 presents the distribution of curated images across their respective groups, with a breakdown of group sizes. Each group comprises consecutive pages. Specifically, there are 57 individual pages, 86 groups containing 2 consecutive pages, 6 groups with 3 consecutive pages, and a singular group of 4 consecutive pages. When partitioning, the integrity of these groups was maintained, ensuring no group was split.

A. Datasets

Table A.6: Number of page groups per partition

# Pages	1	2	3	4
Train	42	65	6	0
Test	15	21	0	1
Total	57	86	6	1

Within Figure A.8, the act distribution across a two-page group is depicted. The sequence of parts of acts “FIFI” is discernible, representing one full act (notated as “IF”) and two partial acts. The former entails the conclusion of an act (“F”) on the first page, presumably having its onset on preceding pages that aren’t incorporated in this group, thus remain untagged. The latter showcases the commencement of an act (“I”), anticipated to culminate on pages that follow.



Figure A.8: In a two-page group, one can observe the “FIFI” sequence. This sequence represents one full act (denoted as “IF”) alongside two fragmented acts.

List of Figures

1	Dependency diagram between the chapters	xvi
2.1	Handwritten number in grayscale represented as a matrix.	6
2.2	Convolution diagram.	17
2.3	Faster R-CNN in a single network.	20
2.4	Pixel image neighborhood vs. Graph node neighborhood.	21
2.5	The Transformer - model architecture.	26
3.1	Two examples from the Archivo General de Indias. Severe degradation can be observed	30
3.2	A notarial deed composed by two pages from the JMBD4950 book, AHPC dataset.	32
3.3	Two left pages from the Hisclima dataset.	33
3.4	Example of an act of six consecutive pages from the JMBD4950 corpus from AHPC.	36
3.5	Example of two consecutive pages from the Nesle corpus .	37
3.6	Example of two consecutive pages from the Denis corpus	37
3.7	Page 78 from the folder JMBD4950.	39
3.8	A notarial deed of class “Power of Attorney”.	40
3.9	Challenges encountered in the Jeannette and Albatross datasets.	45
4.1	Notation act example.	51
4.2	Topology of the Consistency Constraints HMM.	55
4.3	General model solution for act segmentation	56
4.4	Transformation process from act to bag of words.	58
4.5	Markov chain of the Consistency Constraints for AHPC. .	61
4.6	Illustration of the proposed comprehensive pipeline for AHPC.	64
4.7	Missclassified pages 717 and 888 from JMBD4952 book	69

List of Figures

4.8	Learning curve when training with each bundle using ResNet50.	70
4.9	Examples of synthetic data for Nesle, Denis, and Navarre.	74
4.10	Illustration of the proposed comprehensive pipeline with “IMFC” tagging.	77
4.11	Results from a page of Nesle.	90
4.12	Results from a page of Denis.	91
4.13	Markov Chain for the Simancas Archive	94
4.14	PARTs from acts from Simancas Search	94
5.1	Leaving-one-out classification error rate for CSC and OSC	113
5.2	Leaving-one-out classification error rate for CSC using headers.	116
5.3	Rejection performance for bMLP-2	118
6.1	Example of IE query-value	123
6.2	Example of sub-structures	124
6.3	Example of detecting sub-structures from graphs	126
6.4	Example of extracted paths	127
6.5	General pipeline for IE	128
6.6	Automatically detected textlines at the left with MaskRCNN and reference textlines at the right.	132
6.7	GNN for structure detection pipeline	134
6.8	GNN image results for structures	135
A.1	Example of two consecutive pages from the Nesle corpus.	156
A.2	Example of two consecutive pages from the Denis corpus.	157
A.3	Example of two consecutive pages from the Navarre corpus.	158
A.4	Three examples of headers for the books JMBD4949 and JMBD4950.	160
A.5	Three table examples display Jeannette and Albatross.	163
A.6	Examples of challenges in the Jeannette and Albatross datasets	164
A.7	Examples of Simancas Archive	167
A.8	Examples of two tagged pages for the Simancas Archive.	168

List of Tables

4.1	Cross-entropy results for AHPC segmentation	67
4.2	Achieved BSER and CAER results by different decoders in AHPC segmentation	69
4.3	CER and WER results using the DAN model	82
4.4	Average Precision for RPN results without classes.	83
4.5	mAP for RPN results with classes and unconstrained output.	83
4.6	mAP for RPN results after CCs.	84
4.7	CAER results using RPN + DAN and the unconstrained decoder.	85
4.8	CAER results using RPN + DAN and a greedy decoder.	86
4.9	CAER results using RPN + DAN and a viterbi-based decoder.	87
4.10	CAER results using end-to-end DAN.	88
4.11	Simancas Archive Act Classification	95
4.12	mAPs for the RPN detection in the Simancas Corpus	97
4.13	CAER (%) for the Simancas Corpus after using Viterbi decoding.	98
5.1	Number of documents and RWs for JMBD4949 and JMBD4950.	112
5.2	Classification error rate of threshold-less methods	114
5.3	Confusion matrix for PrIx MLP-2 OSC with $n = 2048$	115
5.4	OSC classification + rejection for bMLP-2	117
6.1	Text recognition results for the HisClima Corpus	133
6.2	F_1 for structure recognition	134
6.3	Classification Error Rate of header classification.	137
6.4	Classification Error Rate of span edges.	138
6.5	Information extraction F_1 results for the HisClima corpus	141

List of Tables

A.1	Number of pages and acts in the Nesle, Denis, and Navarre folders.	156
A.2	Statistics for the bundles JMBD4946, JMBD4949, JMBD4950 and JMBD4952.	160
A.3	Statistics of document classes for JMBD4949 and JMBD4950	161
A.4	Statistics of the Jeannette and Albatross corpus	165
A.5	The number of acts in the Simancas Archive and their distribution.	167
A.6	Number of page groups per partition	168

Bibliography

- [Adi+19] Adiga, D. et al. “Table Structure Recognition Based on Cell Relationship, a Bottom-Up Approach”. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. Varna, Bulgaria: INCOMA Ltd., Sept. 2019, pp. 1–8.
- [Aga18] Agarap, A. F. “Deep learning using rectified linear units (relu)”. In: *arXiv preprint arXiv:1803.08375* (2018).
- [Aiz03] Aizawa, A. “An information-theoretic perspective of tf–idf measures”. In: *Inf. Proc. & Management* vol. 39, no. 1 (2003), pp. 45–65.
- [AK15] Awad, M. and Khanna, R. “Support Vector Machines for Classification”. In: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkeley, CA: Apress, 2015, pp. 39–66.
- [Alz+21] Alzubaidi, L. et al. “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”. In: *Journal of Big Data* vol. 8, no. 1 (Mar. 2021), p. 53.
- [And+22] Andrés, J. et al. “Information Extraction from Handwritten Tables in Historical Documents”. In: *Document Analysis Systems*. Ed. by Uchida, S., Barney, E., and Eglin, V. Cham: Springer International Publishing, 2022, pp. 184–198.
- [AZ12] Aggarwal, C. C. and Zhai, C. *Mining text data*. Springer Science & Business Media, 2012.
- [Bat+18] Battaglia, P. W. et al. “Relational inductive biases, deep learning, and graph networks”. In: *CoRR* vol. abs/1806.01261 (2018). arXiv: 1806.01261.

- [BB07] Bottou, L. and Bousquet, O. “The Tradeoffs of Large Scale Learning”. In: *Proceedings of the 20th International Conference on Neural Information Processing Systems*. NIPS’07. Vancouver, British Columbia, Canada: Curran Associates Inc., 2007, pp. 161–168.
- [Bis+21] Biswas, S. et al. “Beyond document object detection: instance-level segmentation of complex layouts”. In: vol. 24. Springer Science and Business Media Deutschland GmbH, Sept. 2021, pp. 269–281.
- [Bis07] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 1st ed. Springer, 2007.
- [BKP20] Boillet, M., Kermorvant, C., and Paquet, T. “Multiple document datasets pre-training improves text line detection with deep neural networks”. In: International Conference on Pattern Recognition (ICPR), 2020, pp. 2134–2141.
- [BKP22] Boillet, M., Kermorvant, C., and Paquet, T. “Robust text line detection in historical documents: learning and evaluation methods”. In: *International Journal on Document Analysis and Recognition* vol. 25 (2 June 2022), pp. 95–114.
- [Ble+23] Blecher, L. et al. *Nougat: Neural Optical Understanding for Academic Documents*. 2023. arXiv: 2308.13418 [cs.LG].
- [BLM16] Bluche, T., Louradour, J., and Messina, R. “Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention”. In: Apr. 2016.
- [Blu+17] Bluche, T. et al. “Preparatory KWS Experiments for Large-Scale Indexing of a Vast Medieval Manuscript Collection in the HIMANIS Project”. In: *14th ICDAR*. Vol. 01. Nov. 2017, pp. 311–316.
- [Blu16] Bluche, T. “Joint Line Segmentation and Transcription for End-to-End Handwritten Paragraph Recognition”. In: 2016, pp. 838–846.
- [Bro+21] Bronstein, M. M. et al. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. 2021. arXiv: 2104.13478 [cs.LG].

- [Cam20] Campos, V. B. “Advances in document layout analysis”. PhD thesis. Universitat Politècnica de València, 2020.
- [CB07] Chen, N. and Blostein, D. “A survey of document image classification: problem statement, classifier architecture and performance evaluation”. In: *International Journal of Document Analysis and Recognition (IJDAR)* vol. 10, no. 1 (2007), pp. 1–16.
- [CCP22] Coquenot, D., Chatelain, C., and Paquet, T. “End-to-end Handwritten Paragraph Text Recognition Using a Vertical Attention Network”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Jan. 2022).
- [CCP23] Coquenot, D., Chatelain, C., and Paquet, T. “DAN: a Segmentation-free Document Attention Network for Handwritten Document Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), pp. 1–17.
- [CG22] Chambers, L. and Gaber, M. M. “DeepStreamOS: Fast open-Set classification for convolutional neural networks”. In: *Pattern Recognition Letters* vol. 154 (2022), pp. 75–82.
- [Chi+19] Chiang, W.-L. et al. “Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 257–266.
- [Con+22] Constum, T. et al. “Recognition and Information Extraction in Historical Handwritten Tables: Toward Understanding Early 20th Century Paris Census”. In: *Document Analysis Systems*. Cham: Springer International Publishing, 2022, pp. 143–157.
- [DBV16] Defferrard, M., Bresson, X., and Vandergheynst, P. “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”. In: *Advances in Neural Information Processing Systems*. 2016.

- [Det+22] Dettmers, T. et al. “GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale”. In: *Advances in Neural Information Processing Systems*. Ed. by Oh, A. H. et al. 2022.
- [Dev+19] Devlin, J. et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *North American Chapter of the Association for Computational Linguistics*. 2019.
- [DH+73] Duda, R. O., Hart, P. E., et al. *Pattern classification and scene analysis*. Vol. 3. Wiley New York, 1973.
- [DKM11] Doucet, A., Kazai, G., and Meunier, J.-L. “ICDAR 2011 Book Structure Extraction Competition”. In: *2011 International Conference on Document Analysis and Recognition*. 2011, pp. 1501–1505.
- [DM19] Déjean, H. and Meunier, J.-L. “Table Rows Segmentation”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. 2019, pp. 461–466.
- [Dom14] Dominiczak, M. H. “The Aesthetics of Texts: Medieval Illuminated Manuscripts”. In: *Clinical Chemistry* vol. 60, no. 6 (June 2014), pp. 907–908.
- [Dou+09] Doucet, A. et al. “ICDAR 2009 Book Structure Extraction Competition”. In: *2009 10th International Conference on Document Analysis and Recognition*. 2009, pp. 1408–1412.
- [DV18] Dumoulin, V. and Visin, F. *A guide to convolution arithmetic for deep learning*. 2018. arXiv: 1603.07285 [stat.ML].
- [DZ23] Dettmers, T. and Zettlemoyer, L. *The case for 4-bit precision: k-bit Inference Scaling Laws*. 2023. arXiv: 2212.09720 [cs.LG].
- [Flo+22] Flores, J. J. et al. “Classification of untranscribed handwritten notarial documents by textual contents”. In: *Pattern Recognition and Image Analysis: 10th Iberian Conference, IbPRIA 2022, Aveiro, Portugal, May 4–6, 2022, Proceedings*. Springer. 2022, pp. 14–26.

- [FM82] Fukushima, K. and Miyake, S. “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition”. In: *Competition and Cooperation in Neural Nets*. Ed. by Amari, S.-i. and Arbib, M. A. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982, pp. 267–285.
- [FSV02] Frasconi, P., Soda, G., and Vullo, A. “Hidden Markov Models for Text Categorization in Multi-Page Documents”. In: *J. Intell. Inf. Syst.* vol. 18, no. 2–3 (Mar. 2002), pp. 195–217.
- [Gao+19] Gao, L. et al. “ICDAR 2019 competition on table detection and recognition (cTDaR)”. In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR (2019)*, pp. 1510–1515.
- [GB10] Glorot, X. and Bengio, Y. “Understanding the difficulty of training deep feedforward neural networks”. In: *Journal of Machine Learning Research* vol. 9 (2010), pp. 249–256.
- [GBC16] Goodfellow, I. J., Bengio, Y., and Courville, A. *Deep Learning*. <http://www.deeplearningbook.org>. Cambridge, MA, USA: MIT Press, 2016.
- [GHC21] Geng, C., Huang, S.-J., and Chen, S. “Recent Advances in Open Set Recognition: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 43, no. 10 (2021), pp. 3614–3631.
- [Gil+17a] Gilani, A. et al. “Table Detection Using Deep Learning”. In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR* vol. 1 (2017), pp. 771–776.
- [Gil+17b] Gilmer, J. et al. “Neural Message Passing for Quantum Chemistry”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Precup, D. and Teh, Y. W. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1263–1272.
- [Gil96] Giles, M. W. “From Gutenberg to Gigabytes: Scholarly Communication in the Age of Cyberspace”. In: *The Journal of Politics* vol. 58, no. 3 (1996), pp. 613–626.

- [Gra+23] Granell, E. et al. “Processing a large collection of historical tabular images”. In: *Pattern Recognition Letters* vol. 170 (2023), pp. 9–16.
- [Grü+17] Grüning, T. et al. *READ-BAD: A New Dataset and Evaluation Scheme for Baseline Detection in Archival Documents*. 2017. arXiv: 1705.03311 [cs.CV].
- [GS08] Graves, A. and Schmidhuber, J. “Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks”. In: *Proceedings of the 21st International Conference on Neural Information Processing Systems*. NIPS’08. Vancouver, British Columbia, Canada: Curran Associates Inc., 2008, pp. 545–552.
- [He+15] He, K. et al. *Deep Residual Learning for Image Recognition*. 2015.
- [He+17] He, K. et al. “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988.
- [Hof+22] Hoffmann, J. et al. *Training Compute-Optimal Large Language Models*. 2022. arXiv: 2203.15556 [cs.CL].
- [HSW89] Hornik, K., Stinchcombe, M., and White, H. “Multilayer feed-forward networks are universal approximators”. In: *Neural Networks* vol. 2, no. 5 (1989), pp. 359–366.
- [Hua+22] Huang, H. et al. “Class-Specific Semantic Reconstruction for Open Set Recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* vol. PP (Aug. 2022).
- [HYL17] Hamilton, W., Ying, Z., and Leskovec, J. “Inductive Representation Learning on Large Graphs”. In: *Advances in Neural Information Processing Systems*. Ed. by Guyon, I. et al. Vol. 30. Curran Associates, Inc., 2017.
- [IKT05] Ikonomakis, M., Kotsiantis, S., and Tampakas, V. “Text classification using machine learning techniques.” In: *WSEAS transactions on computers* vol. 4,8 (2005), pp. 966–974.

- [Isl+23] Islam, S. et al. *A Comprehensive Survey on Applications of Transformers for Deep Learning Tasks*. 2023. arXiv: 2306.07303 [cs.LG].
- [Jia+23] Jiang, A. Q. et al. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL].
- [Joa96] Joachims, T. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. Tech. rep. Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.
- [Kan+14] Kang, L. et al. “Convolutional neural networks for document image classification”. In: *2014 22nd International Conference on Pattern Recognition*. IEEE. 2014, pp. 3168–3172.
- [KGC17] Kukacka, J., Golkov, V., and Cremers, D. “Regularization for Deep Learning: A Taxonomy”. In: *CoRR* vol. abs/1710.10686 (2017). arXiv: 1710.10686.
- [Kha+10] Khan, A. et al. “A review of machine learning algorithms for text-documents classification”. In: *Journal of advances in information technology* vol. 1, no. 1 (2010), pp. 4–20.
- [Kim+22] Kim, G. et al. “OCR-free Document Understanding Transformer”. In: Nov. 2022.
- [KKJ12] Kumar, S., Khan, Z., and Jain, A. “A review of content based image classification using machine learning approach”. In: *International Journal of Advanced Computer Research* vol. 2, no. 3 (2012), p. 55.
- [Kuh55] Kuhn, H. W. “The Hungarian method for the assignment problem”. In: *Naval Research Logistics Quarterly* vol. 2, no. 1-2 (1955), pp. 83–97.
- [KW17] Kipf, T. N. and Welling, M. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. ICLR '17. Palais des Congrès Neptune, Toulon, France, 2017.

- [Lan+18] Lang, E. et al. “Probabilistic Indexing and Search for Information Extraction on Handwritten German Parish Records”. In: *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 2018, pp. 44–49.
- [Lec+89] Lecun, Y. et al. “Backpropagation applied to handwritten zip code recognition”. English (US). In: *Neural Computation* vol. 1, no. 4 (1989), pp. 541–551.
- [Lec+90] Lecun, Y. et al. “Handwritten digit recognition with a backpropagation network”. English (US). In: *Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO*. Ed. by Touretzky, D. Vol. 2. Morgan Kaufmann, 1990.
- [LH17] Loshchilov, I. and Hutter, F. *Decoupled Weight Decay Regularization*. 2017.
- [Li+22a] Li, X.-H. et al. “Table Structure Recognition and Form Parsing by End-to-End Object Detection and Relation Parsing”. In: *Pattern Recognition* vol. 132 (2022), p. 108946.
- [Li+22b] Li, Z. et al. “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects”. In: *IEEE Transactions on Neural Networks and Learning Systems* vol. 33, no. 12 (2022), pp. 6999–7019.
- [Lin+11] Lin, Y. et al. “Large-scale image classification: Fast feature extraction and SVM training”. In: *CVPR 2011*. IEEE. 2011, pp. 1689–1696.
- [Lin+15] Lin, T.-Y. et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].
- [Liu+21] Liu, Z. et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *Proceedings of the IEEE/CVF ICCV*. 2021.
- [Liu+22] Liu, Z. et al. “A ConvNet for the 2020s”. In: *Proceedings of the IEEE/CVF CVPR (2022)*.
- [LK22] Leo, J. and Kalita, J. “Incremental Deep Neural Network Learning Using Classification Confidence Thresholding”. In: *IEEE Transactions on Neural Networks and Learning Systems* vol. 33, no. 12 (2022), pp. 7706–7716.

- [LPR22] Lopez-Lopez, E., Pardo, X. M., and Regueiro, C. V. “Incremental Learning from Low-labelled Stream Data in Open-Set Video Face Recognition”. In: *Pattern Recognition* vol. 131 (2022), p. 108885.
- [MC21] Mahdavi, A. and Carvalho, M. “A Survey on Open Set Recognition”. In: *2021 IEEE Fourth Int. Conf. on Artificial Intelligence and Knowledge Engineering (AIKE)*. 2021, pp. 37–44.
- [Mis19] Misra, D. “Mish: A self regularized non-monotonic neural activation function”. In: *arXiv preprint arXiv:1908.08681* (2019).
- [MLK20] Martínek, J., Lenc, L., and Král, P. “Building an efficient OCR system for historical documents with little training data”. In: *Neural Computing and Applications* vol. 32, no. 23 (Dec. 2020), pp. 17209–17227.
- [MRS08] Manning, C. D., Raghavan, P., and Schtze, H. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.
- [MWY10] Ma, D., Wu, X., and Yang, H. “Efficient Small Object Detection with an Improved Region Proposal Networks”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), pp. 73–80.
- [NDC17] Nguyen, T.-T.-H., Doucet, A., and Coustaty, M. “Enhancing Table of Contents Extraction by System Aggregation”. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 01. 2017, pp. 242–247.
- [Noc+22] Nockels, J. et al. “Understanding the application of handwritten text recognition technology in heritage contexts: a systematic review of Transkribus in published research”. In: *Archival Science* vol. 22, no. 3 (Sept. 2022), pp. 367–392.
- [OSK18] Oliveira, S. A., Seguin, B., and Kaplan, F. “DhSegment: A generic deep-learning approach for document segmentation”. In: vol. 2018-August. IEEE, Dec. 2018, pp. 7–12.

- [Pae+99] Paek, S. et al. “Integration of visual and text-based approaches for the content labeling and classification of photographs”. In: *Acm sigir*. Vol. 99. Citeseer. 1999, pp. 15–19.
- [Par19] Parkinson, R. B. “115Libraries in Ancient Egypt, c.2600–1600 bce”. In: *Libraries before Alexandria: Ancient Near Eastern Traditions*. Oxford University Press, Nov. 2019.
- [PDM19a] Prasad, A., Déjean, H., and Meunier, J. L. “Versatile layout understanding via conjugate graph”. In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR (2019)*, pp. 287–294.
- [PDM19b] Prasad, A., Déjean, H., and Meunier, J.-L. “Versatile Layout Understanding via Conjugate Graph”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. 2019, pp. 287–294.
- [PLK04] Park, S. B., Lee, J. W., and Kim, S. K. “Content-based image classification using a neural network”. In: *Pattern Recognition Letters* vol. 25, no. 3 (2004), pp. 287–300.
- [PM18] Puigcerver, J. and Mocholí, C. *PyLaia*. <https://github.com/jpuigcerver/PyLaia>. 2018.
- [PP23] Parres, D. and Paredes, R. “Fine-Tuning Vision Encoder–Decoder Transformers for Handwriting Text Recognition on Historical Documents”. In: *Document Analysis and Recognition - ICDAR 2023*. Ed. by Fink, G. A. et al. Cham: Springer Nature Switzerland, 2023, pp. 253–268.
- [Pri+21] Prieto, J. R. et al. “Textual-Content-Based Classification of Bundles of Untranscribed Manuscript Images”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 3162–3169.
- [Pri+23a] Prieto, J. R. et al. “Information extraction in handwritten historical logbooks”. In: *Pattern Recognition Letters* vol. 172 (2023), pp. 128–136.
- [Pri+23b] Prieto, J. R. et al. “Open set classification of untranscribed handwritten text image documents”. In: *Pattern Recognition Letters* vol. 172 (2023), pp. 113–120.

- [PSL22] Press, O., Smith, N., and Lewis, M. “Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation”. In: *International Conference on Learning Representations*. 2022.
- [PSM10] Perronnin, F., Sánchez, J., and Mensink, T. “Improving the fisher kernel for large-scale image classification”. In: *European conference on computer vision*. Springer. 2010, pp. 143–156.
- [Pui18] Puigcerver, J. “A Probabilistic Formulation of Keyword Spotting”. PhD thesis. Univ. Politècnica de València, 2018.
- [PV21] Prieto, J. R. and Vidal, E. “Improved Graph Methods for Table Layout Understanding”. In: *Document Analysis and Recognition – ICDAR 2021*. Ed. by Lladós, J., Lopresti, D., and Uchida, S. Cham: Springer International Publishing, 2021, pp. 507–522.
- [QMS19] Qasim, S. R., Mahmood, H., and Shafait, F. “Rethinking table recognition using graph neural networks”. In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR (2019)*, pp. 142–147.
- [QTV19] Quirós, L., Toselli, A. H., and Vidal, E. “Multi-task Layout Analysis of Handwritten Musical Scores”. In: *Iberian Conference on Pattern Recognition and Image Analysis*. Springer. 2019, pp. 123–134.
- [Que05] Quenet, P. “The Diffusion of the Cuneiform Writing System in Northern Mesopotamia: The Earliest Archaeological Evidence”. In: *Iraq* vol. 67, no. 2 (2005), pp. 31–40.
- [Qui17] Quirós, L. *P2PaLA: Page to PAGE Layout Analysis toolkit*. <https://github.com/lquirosd/P2PaLA>. GitHub repository. 2017.
- [Qui22] Quirós, L. “Layout Analysis for Handwritten Documents. A Probabilistic Machine Learning Approach”. PhD thesis. Universitat Politècnica de València, 2022.

- [Rad+21] Radford, A. et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].
- [Raf+20] Raffel, C. et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* vol. 21, no. 140 (2020), pp. 1–67.
- [Ren+17] Ren, S. et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 39, no. 6 (2017), pp. 1137–1149.
- [RHW86] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. “Learning representations by back-propagating errors”. In: *Nature* vol. 323 (1986), pp. 533–536.
- [Rib+19] Riba, P. et al. “Table Detection in Invoice Documents by Graph Neural Networks”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. 2019, pp. 122–127.
- [Rom+13] Romero, V. et al. “The ESPOSALLES database: An ancient marriage license corpus for off-line handwriting recognition”. In: *Pattern Recognition* vol. 46, no. 6 (2013), pp. 1658–1669.
- [Rom+19a] Romero, V. et al. “Information Extraction in Handwritten Marriage Licenses Books”. In: *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing. HIP '19*. Sydney, NSW, Australia, 2019, pp. 66–71.
- [Rom+19b] Romero, V. et al. “Modern vs diplomatic transcripts for historical handwritten text recognition”. In: *Int. Conf. on Image Analysis and Processing (PatReCH workshop)*. Vol. LCNS 11808. Springer. 2019, pp. 103–114.
- [Ros58] Rosenblatt, F. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* vol. 65 6 (1958), pp. 386–408.

- [RS20] Romero, V. and Sánchez, J. A. “The HisClima database: historical weather logs for automatic transcription and information extraction”. In: *ICPR*. 2020.
- [RS21] Romero, V. and Sánchez, J. A. “The HisClima database: historical weather logs for automatic transcription and information extraction”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. 2021, pp. 10141–10148.
- [Rud17] Ruder, S. “An overview of gradient descent optimization algorithms”. In: vol. 14 (2017), pp. 2–3. eprint: 1609.04747.
- [RW17] Rawat, W. and Wang, Z. “Deep convolutional neural networks for image classification: A comprehensive review”. In: *Neural computation* vol. 29, no. 9 (2017), pp. 2352–2449.
- [San+11] Sande, K. E. A. van de et al. “Segmentation As Selective Search for Object Recognition”. In: *IEEE International Conference on Computer Vision*. 2011.
- [Sán+19] Sánchez, J. A. et al. “A set of benchmarks for Handwritten Text Recognition on historical documents”. In: *Pattern Recognition* vol. 94 (2019), pp. 122–134.
- [SB88] Salton, G. and Buckley, C. “Term-weighting approaches in automatic text retrieval”. In: *Inf. Proc. & Management* vol. 24, no. 5 (1988), pp. 513/523.
- [Sch22] Schmidhuber, J. *Annotated History of Modern AI and Deep Learning*. 2022. arXiv: 2212.11279 [cs.LG].
- [Shu+19] Shu, Y. et al. *P-ODN: Prototype based Open Deep Network for Open Set Recognition*. 2019.
- [Sid+19] Siddiqui, S. A. et al. “DeepTabStR: Deep learning based table structure recognition”. In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR* (2019), pp. 1403–1409.
- [SJB14] Scheirer, W. J., Jain, L. P., and Boulton, T. E. “Probability Models for Open Set Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 36, no. 11 (2014), pp. 2317–2324.

- [SK19] Shorten, C. and Khoshgoftaar, T. M. “A survey on Image Data Augmentation for Deep Learning”. In: *Journal of Big Data* vol. 6 (2019), pp. 1–48.
- [SOE22] Sevim, S., Omurca, S. İ., and Ekinçi, E. “Document Image Classification with Vision Transformers”. In: *Electrical and Computer Engineering*. Ed. by Seyman, M. N. Cham: Springer International Publishing, 2022, pp. 68–81.
- [Sri+14] Srivastava, N. et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* vol. 15, no. 56 (2014), pp. 1929–1958.
- [Sto02] Stolcke, A. “SRILM—an extensible language modeling toolkit.” In: *Proceedings of the 3rd Annual Conference of the International Speech Communication Association (Interspeech)*. 2002, pp. 901–904.
- [SXL17] Shu, L., Xu, H., and Liu, B. “DOC: Deep Open Classification of Text Documents”. In: *Proceedings of the 2017 Conf. on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2911–2916.
- [Tos+16] Toselli, A. H. et al. “HMM Word Graph based Keyword Spotting in Handwritten Document Images”. In: *Information Sciences* vol. 370-371 (2016), pp. 497–518.
- [Tos+19] Toselli, A. H. et al. “Making two vast historical manuscript collections searchable and extracting meaningful textual features through large-scale probabilistic indexing”. In: *Int. Conf. on Document Analysis and Recogn. (ICDAR)*. IEEE. 2019, pp. 108–113.
- [Tou+23] Touvron, H. et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].
- [TV23] Toselli, A. H. and Vidal, E. “Revisiting Bag-of-Word Metrics to Assess End-To-End Text Image Recognition Results”. In: *Preprint*. 2023.

- [TZZ13] Tian, L., Zheng, D., and Zhu, C. “Image classification based on the combination of text features and visual features”. In: *International journal of intelligent systems* vol. 28, no. 3 (2013), pp. 242–256.
- [Vas+17] Vaswani, A. et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].
- [Vel+17] Veličković, P. et al. “Graph Attention Networks”. In: *6th International Conference on Learning Representations* (2017).
- [Vel+19] Veličković, P. et al. “Deep Graph Infomax”. In: *International Conference on Learning Representations*. 2019.
- [Vel23] Veličković, P. *Everything is Connected: Graph Neural Networks*. 2023. arXiv: 2301.08210 [cs.LG].
- [Vid+20] Vidal, E. et al. “The Carabela project and manuscript collection: large-scale probabilistic indexing and content-based classification”. In: *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE. 2020, pp. 85–90.
- [Vid+23] Vidal, E. et al. “End-to-End page-Level assessment of handwritten text recognition”. In: *Pattern Recognition* vol. 142 (2023), p. 109695.
- [VJ01] Viola, P. and Jones, M. “Robust real-time face detection”. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. 2001, pp. 747–747.
- [VTP21] Vidal, E., Toselli, A. H., and Puigcerver, J. “A probabilistic framework for lexicon-based keyword spotting in handwritten text images”. In: *arXiv preprint arXiv:2104.04556* (2021).
- [Wan+19] Wang, Y. et al. “Dynamic Graph CNN for Learning on Point Clouds”. In: *ACM Trans. Graph.* vol. 38, no. 5 (Oct. 2019).
- [Wei+21] Weihong, L. et al. “ViBERTgrid: A Jointly Trained Multi-Modal 2D Document Representation for Key Information Extraction from Documents”. In: *ICDAR*. 2021.

- [WMG13] Wu, Z., Mitra, P., and Giles, C. L. “Table of contents recognition and extraction for heterogeneous book documents”. In: *2013 12th International Conference on Document Analysis and Recognition*. 2013, pp. 1205–1209.
- [Wol+20a] Wolf, T. et al. *HuggingFace’s Transformers: State-of-the-art Natural Language Processing*. 2020. arXiv: 1910.03771 [cs.CL].
- [Wol+20b] Wolf, T. et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Oct. 2020, pp. 38–45.
- [Wu+19a] Wu, Y. et al. *Detectron2*. <https://github.com/facebookresearch/detectron2>. 2019.
- [Wu+19b] Wu, Z. et al. “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* vol. 32 (2019), pp. 4–24.
- [XLW17] Xiong, Y.-J., Lu, Y., and Wang, P. S. “Off-line text-independent writer recognition: A survey”. In: *International Journal of Pattern Recognition and Artificial Intelligence* vol. 31, no. 05 (2017), p. 1756008.
- [Xu+21] Xu, Y. et al. “LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 2579–2591.
- [Yan+22a] Yang, H.-M. et al. “Convolutional Prototype Network for Open Set Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 44, no. 5 (2022), pp. 2358–2370.
- [Yan+22b] Yang, Y. et al. “A Survey of Information Extraction Based on Deep Learning”. In: *Applied Sciences* vol. 12, no. 19 (2022).

- [Yim+21] Yim, M. et al. “SynthTIGER: Synthetic Text Image GENERator Towards Better Text Recognition Models”. In: *International Conference on Document Analysis and Recognition*. Springer. 2021, pp. 109–124.
- [YML19] Yao, L., Mao, C., and Luo, Y. “Graph Convolutional Networks for Text Classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 33, no. 01 (2019), pp. 7370–7377.
- [Yos+19] Yoshihashi, R. et al. “Classification-Reconstruction Learning for Open-Set Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 4016–4025.