NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

AFONSO PIMENTA DE OLIVEIRA E SILVA

Master in Computer Science and Engineering

AUTOMATED DATA EXTRACTION FROM DOCUMENTS USING AI: RESPONSE AUTOMATION AND INTEGRATION IN DOCUMENT MANAGEMENT SYSTEMS

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon

September, 2025

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

AUTOMATED DATA EXTRACTION FROM DOCUMENTS USING AI: RESPONSE AUTOMATION AND INTEGRATION IN DOCUMENT MANAGEMENT SYSTEMS

AFONSO PIMENTA DE OLIVEIRA E SILVA

Master in Computer Science and Engineering

Adviser: Joaquim Silva
Assistant Professor, NOVA University Lisbon

Co-advisers: Adriano Campos
Technical director, Caixa Mágica Software

Miguel Roque
AI Lab Lead, Caixa Mágica Software

Examination Committee

Chair: Name of the committee chairperson
Full Professor, FCT-NOVA

Rapporteur: Name of a rapporteur
Associate Professor, Another University

Adviser: Joaquim Silva
Assistant Professor, NOVA University Lisbon

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
September, 2025

Automated Data Extraction from Documents Using AI: Response Automation and Integration in Document Management Systems

Copyright © Afonso Pimenta de Oliveira e Silva, NOVA School of Science and Technology, NOVA University Lisbon.
The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.3.1) [29].

To my family and friends, for their constant support and encouragement.

Acknowledgements

I would like to express my deepest gratitude to all those who supported and accompanied me throughout the course of this thesis.

First of all, I would like to thank Caixa Mágica Software for the opportunity to carry out this dissertation in collaboration with the company. The trust placed in me, the sharing of knowledge and the working environment provided were essential to the development of this project. My special thanks go to Adriano Campos and Miguel Roque for their constant support, availability and valuable guidance throughout the process.

I would also like to express my deepest gratitude to my family, especially to my parents and siblings, for their unconditional love, continuous support and for always providing me with the necessary conditions to pursue this academic path. Your presence and encouragement have been fundamental in motivating me to go further and further.

To my close friends and university colleagues, thank you for the companionship, shared efforts, and encouragement over the years. The moments of study, work, and friendship we experienced together have made this academic journey both meaningful and memorable.

Lastly, I want to thank myself for the dedication, resilience, and determination I invested in this path. This thesis is also a reflection of my own perseverance and the promise I made never to give up.

”

“Great things are not done by impulse, but by a series of small things brought together.”

— Vincent van Gogh

Abstract

Organizations are increasingly challenged by the need to process large volumes of unstructured documents such as invoices, receipts, and contracts. Traditional rule-based approaches often fail to capture the variability and complexity of such data, leading to inefficiencies and high error rates. This thesis investigates two AI-driven strategies for automated information extraction from unstructured documents. The first combines Optical Character Recognition (OCR) through Tesseract with a Large Language Model (LLaMA) to identify and extract key fields from recognized text. The second employs Donut, a transformer-based vision-language model designed for end-to-end document understanding, which directly maps document images to structured outputs without an intermediate OCR step. A comparative evaluation is conducted across multiple measures like character and word error rates, extraction accuracy, latency, and scalability to assess the trade-offs between modular and end-to-end pipelines. The results highlight the strengths and limitations of each method, providing insights into the practical deployment of intelligent document processing systems. This work contributes to advancing the field by offering a systematic comparison of OCR-LLM pipelines versus transformer-based end-to-end models for real-world unstructured document processing.

Keywords: Artificial Intelligence, Document Analysis, Information Extraction, Optical Character Recognition, Unstructured Data, Natural Language Processing, Computer Vision, Transformer Models, Large Language Models, Multimodal Model

Resumo

As organizações enfrentam desafios crescentes na necessidade de processar grandes volumes de documentos não estruturados, como faturas, recibos e contratos. As abordagens tradicionais baseadas em regras revelam-se frequentemente incapazes de capturar a variabilidade e a complexidade destes dados, resultando em ineficiências e elevadas taxas de erro. Esta dissertação investiga duas estratégias baseadas em inteligência artificial para a extração automática de informação a partir de documentos não estruturados. A primeira combina Reconhecimento Óptico de Caracteres (OCR) através do Tesseract com um Large Language Model (LLaMA) para identificar e extrair campos-chave a partir do texto reconhecido. A segunda recorre ao Donut, um modelo multimodal de visão e linguagem baseado em transformadores, que mapeia diretamente imagens de documentos para saídas estruturadas sem necessidade de uma etapa intermédia de OCR. É realizada uma avaliação comparativa com base em várias métricas, como taxas de erro de caracteres e palavras, latência e escalabilidade, de forma a analisar os compromissos entre pipelines modulares e de ponta a ponta. Os resultados evidenciam os pontos fortes e limitações de cada método, fornecendo perspetivas relevantes para a aplicação prática de sistemas inteligentes de processamento documental. Este trabalho contribui para o avanço da área ao oferecer uma comparação sistemática entre pipelines OCR-LLM e modelos de ponta a ponta baseados em transformadores no processamento de documentos não estruturados em cenários reais.

Palavras-chave: Inteligência Artificial, Análise de Documentos, Extração de Informação, Reconhecimento Óptico de Caracteres, Dados Não Estruturados, Processamento de Linguagem Natural, Visão por Computador, Modelos Transformer, Modelos de Linguagem de Grande Escala, Modelos Multimodais

Contents

List of Figures x

List of Tables xii

1 Introduction 1
- 1.1 Contextualization and Motivation . . . 2
- 1.2 Objectives . . . 3
- 1.3 Main Contributions . . . 3
- 1.4 Document Structure . . . 4

2 State-of-the-Art 6
- 2.1 Overview of Document Processing . . . 6
- 2.2 AI in Document Processing . . . 7
  - 2.2.1 Traditional Machine Learning Techniques in Document Processing 9
  - 2.2.2 Optical Character Recognition (OCR) . . . 9
  - 2.2.3 Natural Language Processing (NLP) . . . 10
  - 2.2.4 Named Entity Recognition (NER) and Relationship Extraction . . 11
  - 2.2.5 Semantic Segmentation . . . 13
  - 2.2.6 Robotic Process Automation . . . 14
  - 2.2.7 Large Language Models (LLMs) . . . 14
  - 2.2.8 Knowledge Graphs . . . 15
  - 2.2.9 Multimodal Models . . . 17
- 2.3 Integration with Document Management Systems and Databases . . . 18
- 2.4 Summary . . . 19

3 Dataset 21
- 3.1 Description . . . 21
  - 3.1.1 Government Contracts . . . 21
  - 3.1.2 Receipts and Invoices . . . 22
- 3.2 Preprocessing . . . 23

3.2.1 Government Contracts . . . 23
3.2.2 Receipts and Invoices . . . 24

4 Proposed Solution **25**
4.1 System Architecture . . . 25
4.2 Technology Stack . . . 27
4.2.1 Optical Character Recognition (OCR) with Tesseract . . . 27
4.2.2 Language Model for Field Extraction with LLaMA 3.1 (8B): . . . 28
4.2.3 OCR-Free Document Parsing with Donut: . . . 29
4.2.4 Summary: . . . 29
4.3 Evaluation Measures . . . 30

5 Methodology **32**
5.1 Model Architectures and Implementation . . . 32
5.1.1 OCR + LLM Pipeline . . . 32
5.1.2 Donut Model . . . 33
5.2 Environment . . . 34

6 Evaluation **35**
6.1 Experimental setup . . . 35
6.2 Invoices and Receipts . . . 36
6.2.1 Whole Sequence Errors . . . 36
6.2.2 Field-Level Errors . . . 37
6.2.3 Normalized Errors (Ignoring Formatting) . . . 40
6.2.4 Hardware Performance Comparison . . . 42
6.2.5 Summary of Evaluation . . . 45
6.3 Government Contracts . . . 45
6.3.1 Donut Limitations on Contracts . . . 46
6.3.2 Potential Solutions for Donut on Contracts . . . 46
6.3.3 OCR + LLM . . . 46
6.3.3.1 Whole Sequence Errors . . . 47
6.3.3.2 Field-Level Errors . . . 48
6.3.3.3 Normalized Errors . . . 49
6.3.3.4 Hardware Performance . . . 49
6.3.4 Evaluation . . . 50

7 Discussion **52**
7.1 Overview . . . 52
7.2 Comparative Analysis of Approaches . . . 52
7.3 Practical Considerations . . . 53
7.4 Error Patterns and Failure Cases . . . 54
7.5 Limitations of the Experimental Evaluation . . . 54

8 Conclusion and Future Work **56**
8.1 Conclusion 56
8.2 Future Work 57
8.2.1 Human-in-the-Loop Model Improvement 57
8.2.2 Retrieval-Augmented Generation 57
8.2.3 Robustness and Error Mitigation 58
8.2.4 Summary 58

Bibliography **59**

List of Figures

1.1 Global volume of unstructured data [14] 1

2.1 Evolution of the techniques used for information extraction from unstructured documents [31] 8
2.2 Transformer Architecture [33] 12
2.3 A convolutional neural network (CNN) encoder–decoder architecture for semantic segmentation [33] 13
2.4 Example of a knowledge graph [3] 16
2.5 Donut Architecture 17
2.6 Unstructured document processing approaches using AI [31]. 19

3.1 Examples of government contracts 22
3.2 Examples of invoices from the two datasets used in this thesis. 23

4.1 Pipeline Architecture diagram of the Donut approach 25
4.2 Pipeline Architecture diagram of the OCR+LLM approach 26

5.1 Example of label smoothing [11] 34

6.1 Whole sequence errors of both approaches 36
6.2 Field-level extraction errors for the Total field. 37
6.3 Field-level extraction errors for the Date field. 38
6.4 Field-level extraction errors for the Company field. 38
6.5 Field-level extraction errors for the Address field. 39
6.6 Normalized errors for the Total field. 40
6.7 Normalized errors for the Date field. 41
6.8 Normalized errors for the Company field. 41
6.9 Normalized errors for the Address field. 42
6.10 Total execution time of both approaches 43
6.11 Scan Time breakdown of OCR+LLM 44
6.12 Memory usage of both approaches 44

6.13 Whole sequence errors of OCR + LLM 47
6.14 Field level errors of OCR + LLM 48
6.15 Normalized errors of OCR + LLM 49
6.16 Scan Time breakdown of OCR+LLM 50

List of Tables

2.1 Comparison of data processing techniques 20

3.1 Relevant fields extracted from a government contract for the dataset. 24
3.2 Example of the annotated format for a sample invoice. 24

6.1 Whole sequence errors (mean ± std) for invoices and receipts. 36
6.2 Field-level errors (mean ± std) on invoices/receipts. 37
6.3 Normalized field-level errors (mean ± std) on invoices/receipts. 40
6.4 Hardware execution times for processing 201 images, including average time per image. 43
6.5 Hardware memory usage for processing 201 images. 43
6.6 Whole sequence errors (mean ± std) using OCR+LLM. 47
6.7 Field-level errors (mean ± std) using OCR+LLM. 48
6.8 Normalized field-level errors (mean ± std) using OCR+LLM. 49
6.9 Average time per scan under CPU-only and GPU-enabled configurations. . 50

1 Introduction

With the digitalization of administrative and business processes, organizations are generating increasing volumes of documents in scanned or PDF form. Unlike traditional databases, which follow strict schemas, these documents are often semi-structured or unstructured. They may contain free-flowing text, tables, and irregular layouts that defy standardized representation. These factors make contracts and other scanned documents particularly difficult to organize, search, and analyze, yet they remain critical sources of information.

This explosion of unstructured data has created serious challenges for many organizations, especially those without the tools or expertise to deal with it properly. According to some estimates, around 80% to 90% of all data generated by companies is unstructured [34]. And that number is only growing: global data is expected to reach 175 zettabytes by 2025, much of it unstructured [20], as shown in Figure 1.1.

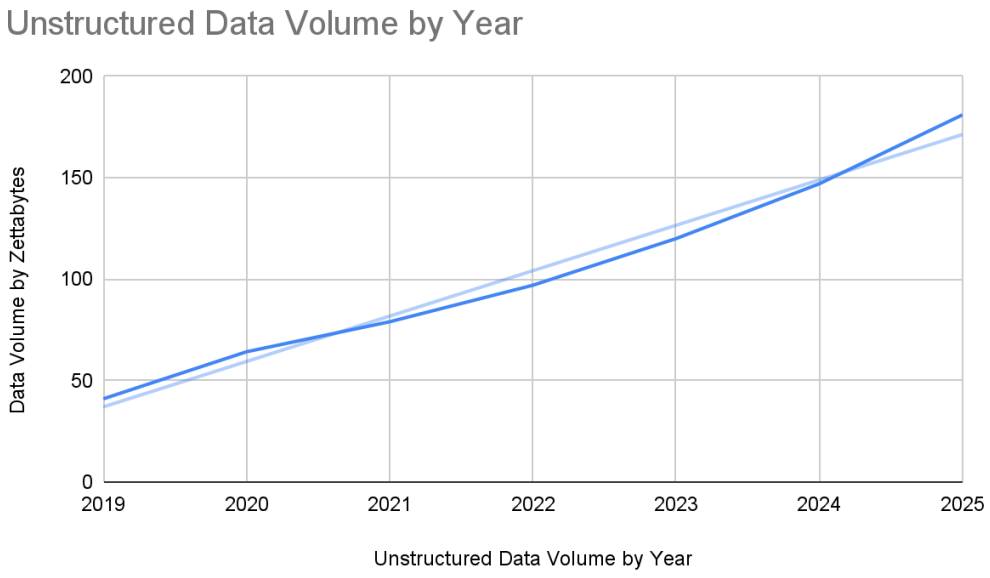


Figure 1.1: Global volume of unstructured data [14]

As this kind of data continues to grow, it is becoming more important than ever to have smart tools that can help organizations make sense of it. When used well, unstructured data can reveal useful patterns, help companies understand what their customers want, and support better business decisions across different industries.

But making use of this data is not easy. It takes time, effort, and money to process unstructured information. Given the rapid pace at which this information is generated, organizations risk overlooking valuable insights unless supported by automated systems capable of processing data at scale.

1.1 Contextualization and Motivation

Organizations today are increasingly overwhelmed by unstructured data in the form of administrative and financial documents. Invoices, receipts, and contracts are particularly common in both the public and private sectors, yet their lack of standardized formatting makes them difficult to process using conventional tools. Layout variability, handwritten annotations, noisy scans, and multilingual content all contribute to the challenge of extracting structured information reliably. Manually reviewing these documents is slow, error-prone, and incapable of keeping pace with operational demands, especially in institutions that handle thousands of documents per month.

This challenge is especially pronounced in public agencies, where budgetary limitations often restrict access to advanced commercial solutions. Many organizations still rely on manual review or outdated rule-based systems, which have poor per when faced with variations in layout or shifts in language and terminology. As a result, significant amounts of administrative effort are spent on repetitive tasks such as verifying invoice totals, extracting contractor information, or validating contract dates.

Recent advances in Artificial Intelligence (AI), particularly in Natural Language Processing (NLP), Machine Learning (ML), and Optical Character Recognition (OCR), have created new opportunities for automating document workflows. End-to-end multimodal models further extend these possibilities by eliminating the need for intermediate OCR, directly mapping document images into structured outputs. However, integrating these techniques into systems that are both scalable and suitable for deployment in resource-constrained environments remains an open challenge.

This thesis addresses this problem by investigating two AI-driven approaches for processing invoices and contracts. The first is a modular pipeline that combines open-source OCR (Tesseract) with a Large Language Model (LLaMA) to extract key fields such as company name, date, address, and total amount. The second employs Donut, a transformer-based vision language model designed for end-to-end document understanding, which directly produces structured outputs from document images. Both approaches are evaluated on publicly available datasets of invoices and receipts, as well as on a custom dataset of work contracts, with a focus on open-source tools, fine-tuned models, and low-resource deployment strategies.

1.2 Objectives

The thesis aims to demonstrate that AI-driven pipelines can effectively automate the extraction of structured information from unstructured financial and administrative documents, and to evaluate the trade-offs between modular OCR+LLM architectures and end-to-end transformer-based models. The specific research objectives are:

- **Design and implement two document-processing approaches:**

  (i) a modular pipeline combining Tesseract OCR with a Large Language Model (LLaMA) for field extraction from recognized text, and (ii) an end-to-end approach using Donut, a vision language transformer that directly maps document images to structured outputs.

- **Prepare and adapt datasets for training and evaluation:**

  With the use of InvoicesReceiptsPT [10] and ICDAR-2019 SROIE [23], as well as the creation of a custom annotated dataset of invoices and work contracts, with fields such as company name, date, address, and total amount.

- **Evaluate system performance using rigorous measures:**

  With a focus on character error rate (CER), word error rate (WER), and field-level accuracy. Latency and scalability are also measured to assess trade-offs between accuracy and efficiency in modular versus end-to-end pipelines.

- **Analyze the practical suitability of each approach**

  For real-world deployment, with emphasis on open-source tools, adaptability, and the constraints of public-sector environments where computational and financial resources may be limited.

By achieving these objectives, the thesis contributes a systematic comparison of OCR-based modular pipelines and end-to-end transformer models for unstructured document processing, offering practical insights for organizations seeking efficient and reliable automation of invoice, receipt, and contract analysis.

1.3 Main Contributions

The main contributions of this thesis can be summarized as follows:

- **Systematic Comparison of Two Paradigms for Document Understanding.**

  This thesis provides a direct evaluations of a modular OCR+LLM pipeline (Tesseract with LLaMA) against an end-to-end multimodal transformer model (Donut) for the task of processing invoices and contracts. The comparison highlights the trade-offs between modular and end-to-end designs in terms of accuracy, latency, and robustness.

- **Creation of Domain-Specific Evaluation Resources.**

  Two publicly available datasets (InvoicesReceiptsPT [10] and ICDAR-2019 SROIE [23]) are adapted for benchmarking, and a manually annotated dataset of Portuguese invoices and contracts is produced. This contribution supplies high-quality, domain-relevant ground truth for evaluating both pipelines and fills a gap in resources for administrative document processing.

- **Development of a Flexible Modular Pipeline.**

  A modular architecture is built that combines open-source OCR and LLMs to extract structured fields such as company name, date, address, and total amount. The design prioritizes adaptability, enabling components to be interchanged or upgraded while remaining suitable for low-resource deployment contexts.

- **Adaptation and Assessment of Donut for Portuguese Administrative Documents.**

  The Donut model is fine-tuned and evaluated on Portuguese invoices and government contracts, demonstrating its ability to bypass OCR and directly generate structured outputs. The fine-tuning process enables the model to adapt to domain-specific characteristics, such as language, formatting variability, and field-level annotation. Its strengths and limitations are systematically analyzed, particularly in comparison with the modular OCR+LLM pipeline for field-level information extraction.

- **Comprehensive Experimental Results and Insights.**

  Extensive experiments using Character Error Rate (CER), Word Error Rate (WER), field-level accuracy, latency, and scalability reveal the relative strengths of each approach. Donut achieves lower latency and streamlined inference, while the modular OCR+LLM pipeline demonstrates greater accuracy and resilience across noisy document inputs.

- **Practical Recommendations for Real-World Adoption.**

  The thesis distills experimental findings into concrete guidelines for intelligent document processing in resource-constrained environments. It clarifies when modular OCR+LLM pipelines remain preferable and when end-to-end transformer models may offer advantages, providing actionable insights for organizations operating under resource constraints.

1.4 Document Structure

The thesis is structured as follows:

- **Chapter 1: Introduction** – Establishes the research context, objectives, main contributions, and provides the document structure.

- **Chapter 2: State-of-the-Art** – Reviews the state of the art in document processing, with emphasis on AI-based techniques such as OCR, NLP, LLMs, multimodal models, and integration with document management systems.
- **Chapter 3: Dataset** – Describes the datasets used in this work and details the preprocessing steps applied.
- **Chapter 4: Proposed Solution** – Presents the system architecture, technology stack, detailed description of both approaches, and the evaluation measures adopted.
- **Chapter 5: Methodology** – Outlines the model architectures and implementation details of both approaches, as well as the experimental environment.
- **Chapter 6: Evaluation** – Provides the experimental setup and results. Includes error analyses on invoices and receipts, hardware performance comparisons, and an evaluation on government contracts.
- **Chapter 7: Discussion** – Interprets the evaluation results, highlighting comparative analysis of the approaches, practical considerations, and observed error patterns.
- **Chapter 8: Conclusion and Future Work** – Summarizes the contributions of this thesis and proposes directions for future research.

2

State-of-the-Art

This chapter provides a critical review of the state-of-the-art in automated document processing, establishing the foundation for the approaches proposed in this thesis. The emphasis is placed on techniques designed to extract structured information from unstructured or semi-structured sources such as invoices, receipts, contracts, emails, and scanned forms.

We begin with a historical perspective, contrasting manual and rule-based approaches with the rise of learning-based methods. This is followed by an examination of core Artificial Intelligence technologies that enable advanced document understanding, including Optical Character Recognition (OCR), Natural Language Processing (NLP), and traditional machine learning techniques. We then analyze specialized strategies for information extraction, such as Named Entity Recognition (NER), knowledge graph construction, semantic segmentation of layouts, and Robotic Process Automation (RPA). The role of Large Language Models (LLMs) and multimodal architectures is given particular attention, as these represent the most recent paradigm shift in document AI.

In addition, we discuss how these techniques are integrated into practical workflows through document management systems and database interfaces, which are critical for real-world adoption. The chapter concludes with a synthesis of insights across methods, highlighting their respective strengths and limitations, and identifying research gaps that motivate the comparative study undertaken in this thesis.

2.1 Overview of Document Processing

Handling documents has always been a central activity in how organizations manage and utilize information, whether the goal is extracting relevant data, classifying content, or storing records for future access. Traditionally, these tasks were carried out manually: employees would carefully read through documents, interpret their content, and then re-enter the necessary information into organizational systems. Although this method was simple and required little technical infrastructure, it was also slow, repetitive, and highly prone to human error, especially when processing large volumes of material under

time pressure.

In an attempt to improve efficiency, many organizations adopted rule-based systems that relied on predefined templates and logic to automate specific steps of the process. These systems provided a meaningful improvement over manual work, particularly for standardized forms, invoices, or fixed layouts. However, their limitations soon became apparent. Rule-based solutions were brittle, struggling with unstructured or semi-structured content such as free-form text, irregularly formatted scanned documents, or handwritten notes. Moreover, they required frequent reconfiguration and maintenance to adapt to new document types or shifts in business processes, making them costly and difficult to scale in the long run.

The past decade has seen a transformative shift with the emergence of artificial intelligence (AI) applied to document processing. Modern systems, powered by machine learning (ML) and natural language processing (NLP), are capable of going beyond rigid rules to learn directly from examples. Often referred to as Intelligent Document Processing (IDP) systems, these solutions adapt to heterogeneous document formats, improve as more data becomes available, and scale more effectively than earlier generations of tools. By leveraging deep learning, they are also able to capture semantic relationships, interpret context, and generalize across multiple domains, offering a degree of flexibility that was previously unattainable [1].

Despite these advances, adopting AI for document processing is not without challenges. Organizations must grapple with issues such as ensuring data privacy and security, collecting sufficient high-quality training data, and managing the technical and computational complexity of large-scale models. Nevertheless, the benefits are compelling: AI-driven systems can substantially enhance efficiency and accuracy, reduce repetitive human labor, and even extract insights from unstructured data sources that would otherwise remain untapped [28].

In short, while traditional approaches to document processing still have their place, AI-driven systems represent a major step forward o"ering smarter, more adaptable tools to handle the growing scale and complexity of today's information.

2.2 AI in Document Processing

Artificial intelligence has progressively transformed how organizations extract value from unstructured documents, moving from rigid, rule-based systems to adaptive, learning-based pipelines. Figure 2.1 illustrates the evolution of techniques used for information extraction from unstructured documents, showing how advances have built upon one another over the past decades.

Early approaches relied on manual processing and handcrafted rules, later advanced by Optical Character Recognition (OCR) in the 1980s, which enabled large-scale digitization of printed text. The subsequent integration of Natural Language Processing (NLP) and machine learning expanded capabilities to entity recognition and classification. Today,

deep learning and transformer-based architectures represent a major leap forward, with multimodal, OCR-free models directly mapping document images to structured outputs.

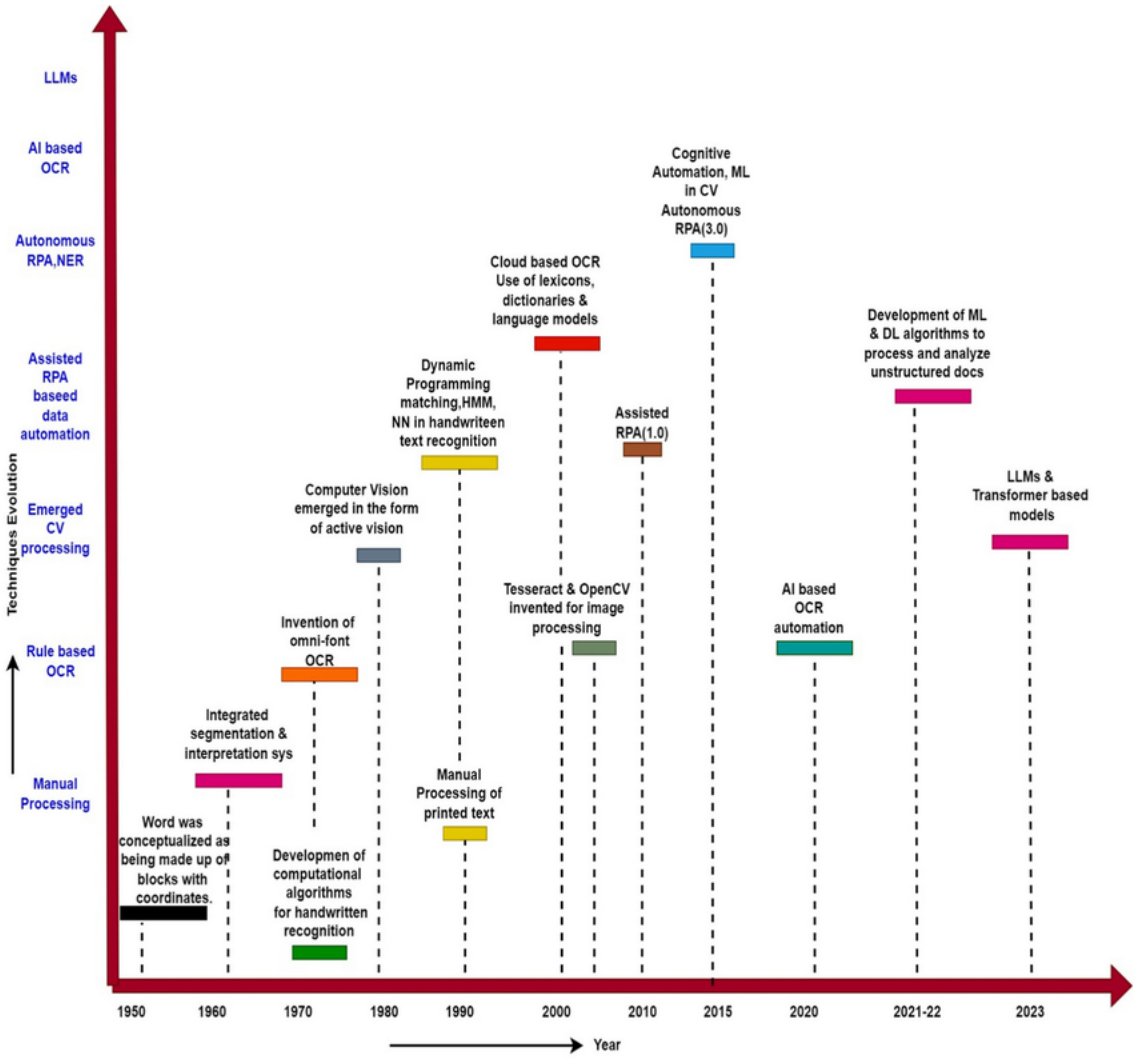


Figure 2.1: Evolution of the techniques used for information extraction from unstructured documents [31]

Artificial intelligence has transformed how organizations extract value from unstructured documents particularly invoices, receipts, and contracts by replacing weak, rule-based systems with learning-based pipelines that generalize across layouts, languages, and scan quality. Architectures typically combine three areas: Optical Character Recognition (OCR) to convert pixels into text, Natural Language Processing (NLP) to derive entities and relations and Machine Learning (ML) to classify, route, validate, and calibrate outputs at scale. Each of these has seen rapid advancements in recent years, contributing to more powerful and versatile document analysis tools. A newer class of multimodal, OCR-free models collapses these stages into end-to-end systems that map document images directly to structured fields, reducing latency and error propagation like Donut [27]

2.2.1 Traditional Machine Learning Techniques in Document Processing

Beyond the specific realms of NLP and OCR, a variety of machine learning approaches contribute to document processing. Machine learning offers adaptability models can be trained on example data to perform classification, clustering, anomaly detection, and more, without explicitly programmed rules.

- **Unsupervised Learning: Making Sense of Unstructured Data:**

  Unsupervised learning methods, such as clustering and anomaly detection, help organize large volumes of unstructured documents without needing labeled training data. For example:

  - Techniques like clustering can group similar documents together without predefined labels [9]. This is useful for discovering latent groupings in large archives of text for instance, clustering might reveal that documents naturally fall into clusters [38] about "financial reports", "customer complaints", "technical specifications", etc., even if those categories weren't explicitly given, such automatic grouping can aid in indexing and search.
  - Anomaly detection can identify documents that don't fit known patterns for example, an outlier in an invoice dataset might indicate a scanning error or a fraudulent document that needs human review. [1].

- **Supervised Learning: Classifying Documents Automatically:**

  In supervised learning, models are trained on labeled datasets to classify documents into predefined categories such as invoices, receipts, or contracts. This approach involves learning patterns and features from the training data that are indicative of each category. For instance, specific keywords, phrases, or layout features within a document can serve as distinguishing features. Once trained, the model can predict the category of new, unseen documents with a high degree of accuracy. This method enhances the efficiency of document management systems by automating the organization and retrieval processes [15].

2.2.2 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is the technology that converts images of text (such as scanned paper documents or photos of receipts) into actual text data that computers can process. OCR has been around for decades, but AI has vastly improved its accuracy and capabilities. Early OCR systems relied on pattern matching for each character and struggled with anything beyond clean, typewritten text. Today's OCR engines incorporate deep learning and computer vision, making them far more robust. Tesseract, a widely used open-source OCR engine, was upgraded to use LSTM neural networks (a form of deep learning) in its recognition pipeline, which significantly boosted its accuracy and

language coverage [40]. Modern OCR can handle a wide variety of fonts, sizes, and even moderate image noise or distortion[37].

Beyond plain text recognition, commercial cloud-based OCR services (such as Google Cloud Vision API, Microsoft Azure OCR, and Amazon Textract) extend OCR to structured outputs [21]. These services not only transcribe text but can also detect document structure like paragraphs, tables, form fields, and checkboxes. For example, Amazon Textract can identify that a piece of text is part of a table cell or that a group of words is a form label versus a form answer [4].

Key AI-driven advancements in OCR include the use of convolutional neural networks (CNNs) for visual feature extraction [46] [47]. CNNs allow OCR systems to better handle variations in visual appearance, effectively learning what characters look like across different contexts rather than relying on hand-tuned filters. This has led to dramatic improvements in reading low-quality scans, rotated or skewed text, and even certain types of handwriting [30]. As a result, OCR is no longer a major bottleneck in many document processing tasks it is fast and sufficiently accurate for printed text under a wide range of conditions.

The classical OCR pipeline typically involves several stages:

- **Image Preprocessing:** Binarization, noise removal, skew correction.
- **Layout Analysis:** Page segmentation, text line detection
- **Character Recognition:** Feature extraction and classification
- **Post-processing:** Language model correction, confidence scoring

However, standard OCR still has limitations. It typically outputs a plain text stream and does not inherently understand higher-level structures or meaning. If a document is complex (e.g., multi-column layouts, mixed text and images, or inconsistent formatting), OCR might capture all the text but jumble the reading order or lose the association between labels and values.

2.2.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) enables computers to interpret and generate human language. Within document processing, NLP techniques allow systems not only to read text but also to understand it at some level identifying which parts are important, what they refer to, and the relationships between them. A pivotal development in NLP has been the rise of transformer-based models, such as BERT [13] and GPT [25]. BERT (Bidirectional Encoder Representations from Transformers) and the GPT family (Generative Pre-trained Transformers) have dramatically improved the accuracy of tasks like entity recognition, text classification, and summarization. These models are pre-trained on massive text datasets and capture rich linguistic patterns. For example, using a model like BERT,

a system can recognize that "John Smith" is a person's name and "Acme Corp" is an organization in an email, even if it has never seen that exact text before it has learned from context how names and organizations typically look. Likewise, named entity recognition (NER) algorithms can tag phrases as dates, monetary amounts, addresses, etc., providing structured data from raw text.

Another powerful capability of modern NLP is understanding relationships and context. Transformer-based models consider the entire context of a word in a sentence (due to the self-attention mechanism in the transformer architecture), so they can resolve references and ambiguities much better than earlier bag-of-words or keyword-based methods. For instance, they can infer that in the sentence "Please send it to John Smith at Acme Corp by March 15," the pronoun "it" likely refers to some document or item previously mentioned, "John Smith" is the recipient, "Acme Corp" is the organization he belongs to, and "March 15" is a date (deadline). Extracting such fine-grained information is crucial for automating workflows and understanding the semantics of documents.

2.2.4 Named Entity Recognition (NER) and Relationship Extraction

Named Entity Recognition identifies specific pieces of information in text such as names of people, dates, organizations, monetary amounts, and so on. Relationship extraction goes a step further by finding meaningful connections between those entities (for example, who works at which company, or which date corresponds to which event). These techniques transform unstructured text into structured data that is easier to analyze and query.

Modern models have greatly improved NER and relationship extraction. BERT-based models introduced by Devlin et al [13], in particular, are highly effective at these tasks. BERT processes language bidirectionally, considering context from both before and after each word in a sentence [48], which helps it disambiguate entities and interpret nuance. For instance, BERT can tell that "Paris" in one context is a city but in another context could be a person's name or something else, by looking at surrounding words. GPT models, on the other hand, are designed to generate fluent, natural text based on a given input [6]. While GPT models are often used for text generation, their understanding is also strong enough to perform extraction when guided appropriately. In fact, GPT-3 and GPT-4 have demonstrated the ability to extract structured information from unstructured text [8], making it valuable for tasks like document summarization and knowledge

Despite their similarities, BERT and GPT have fundamental differences in architecture, training objectives, and application domains. BERT, utilizes an encoder-only transformer architecture, processing text bidirectionally. This means it considers context from both preceding and following words in a sentence, allowing for a comprehensive understanding of language nuances. GPT, employs a decoder-only transformer architecture, processing text unidirectionally (typically left-to-right). This design is optimized for generating coherent and contextually relevant text sequences.

Figure 2.2 illustrates the general Transformer architecture, which consists of two main components:

- **The left side represents the Encoder**, which is primarily used in BERT and similar models for text understanding tasks.
- **The right side represents the Decoder**, which is employed in GPT and other generative models to predict the next token in a sequence.

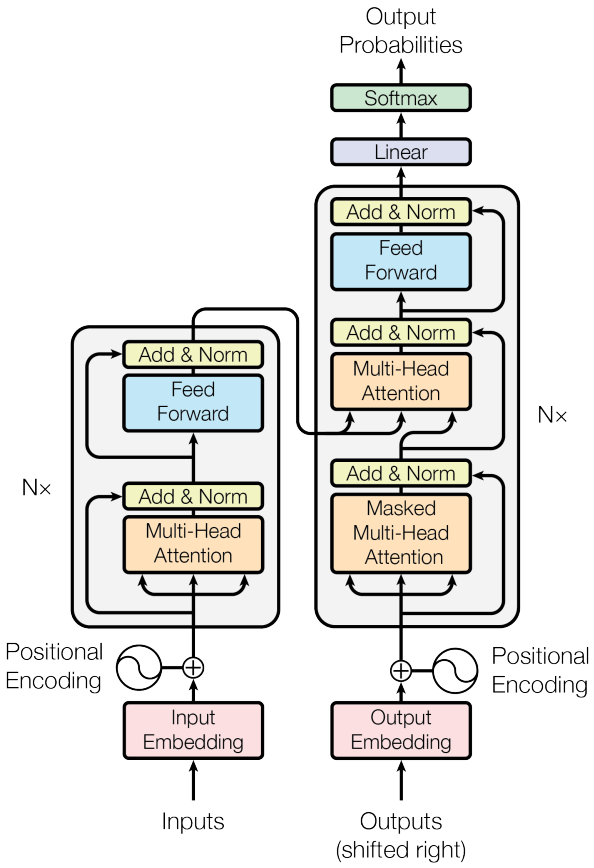


Figure 2.2: Transformer Architecture [33]

In this architecture, the encoder (left) processes the entire input sequence simultaneously, utilizing self-attention mechanisms and feed-forward layers to capture contextual relationships. The decoder (right) takes previously generated tokens as input and incorporates a masked self-attention mechanism, ensuring that predictions for a given token do not depend on future tokens. By leveraging these architectural differences, BERT and GPT are optimized for distinct NLP tasks. BERT excels at understanding and classification, while GPT is specialized in generation and prediction.

The integration of BERT, GPT, and NER-based models has significantly improved the capabilities of NLP systems in handling unstructured text. Their ability to recognize entities and extract relationships has made document processing more accurate and

contextually aware, enabling diverse applications in information retrieval, knowledge graphs, and automated document management.

2.2.5 Semantic Segmentation

Semantic segmentation is a computer vision technique that assigns a class label to each pixel in an image, effectively partitioning the image into meaningful regions. As illustrated in Figure 2.3, a convolutional neural network (CNN) encoder–decoder architecture takes an input RGB image, processes it through layers of convolution, pooling, and upsampling, and produces an output segmentation map in which every region is assigned a semantic category. In the example, pixels corresponding to pedestrians, vehicles, and buildings are separated into distinct classes, making the structure of the scene immediately interpretable.

In the context of document processing, the same principle applies: instead of streets, cars, and people, the categories correspond to layout elements such as titles, paragraphs, tables, images, and captions. This pixel-level understanding of document structure enables more accurate information extraction while preserving the visual and logical layout of the page for instance, ensuring that text within a table is not confused with a caption or body text.

Training robust segmentation models requires large, annotated datasets. Resources such as PubLayNet [51], which provides hundreds of thousands of labeled document images, and DocLayNet [35], which offers detailed human annotations for document layout understanding, have become essential benchmarks. By leveraging these datasets, models learn to maintain the logical and visual structure of documents, which in turn improves the performance of downstream NLP tasks. In summary, semantic segmentation whether applied to street scenes or document layouts plays a crucial role in preserving two-dimensional structure that would otherwise be lost with plain text extraction.

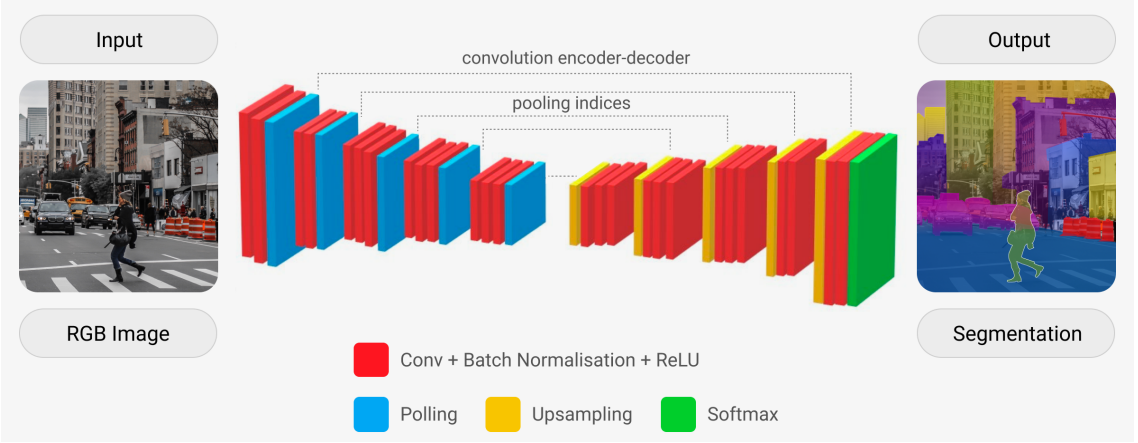


Figure 2.3: A convolutional neural network (CNN) encoder–decoder architecture for semantic segmentation [33]

2.2.6 Robotic Process Automation

Robotic Process Automation (RPA) is widely used to automate repetitive, rule-based tasks such as copying data from one system to another by mimicking user interactions with software. In document processing scenarios, RPA can reliably handle structured digital documents (for instance, extracting specific fields from standardized forms). However, when faced with more complex inputs like free-form emails or scanned PDFs that do not follow a predetermined template, RPA often fails or becomes brittle.

To address this, modern systems fuse RPA with optical character recognition (OCR) and natural language processing (NLP): OCR converts images into text, and NLP techniques interpret that text to extract meaning, allowing RPA to operate on semi-structured or unstructured content [41]. This evolution in approach is embodied in Intelligent Process Automation (IPA), which combines RPA, AI, and ML to dynamically adapt to new document formats, variations in language, and evolving workflows. In IPA architectures, the system may gradually learn to recognize new templates, resolve inconsistencies, and improve extraction accuracy over time, thus overcoming the rigidity of conventional RPA systems [12].

2.2.7 Large Language Models (LLMs)

Large Language Models have brought a huge leap forward in how we process and understand language. These models built on deep learning and the transformer architecture can perform a wide range of tasks such as summarizing documents, translating languages, answering questions, and analyzing sentiment. Their ability to handle massive amounts of text and learn from diverse data makes them especially effective for working with unstructured content.

- **Transformer Architecture:**

  The success of LLMs is largely attributed to the transformer architecture, introduced by Vaswani et al. [44]. Transformers use self-attention mechanisms to process input text in parallel, capturing long-range dependencies and contextual relationships between words. This architecture has become the foundation for state-of-the-art LLMs, enabling models to handle complex language tasks with remarkable efficiency. Notably, for document understanding tasks, transformer models can jointly process textual content and other modalities or metadata. For example, transformers can be extended to also consider visual appearance and spatial layout information in documents, not just linear text. This opens the door to models that truly understand documents in a human-like way, by taking into account how information is presented on the page.

- **LLaMA:**

  Developed by Meta AI, LLaMA is a family of LLMs designed to offer high performance with relatively smaller, efficient models. LLaMA models (e.g., with 7B or 13B parameters) have demonstrated strong capabilities in tasks like summarization and question answering, even when running on limited computing resources [43]. This makes them attractive for deployment in environments where GPU memory or computational power is constrained, such as on-premises servers or edge devices in a secure government setting.

- **DeepSeek:**

  DeepSeek is an advanced LLM developed by DeepSeek AI. Built for real-time applications. It specializes in domain-specific tasks such as entity recognition, document summarization, and context-aware text generation, particularly in specialized fields like healthcare or law. While not a broadly published model like LLaMA or GPT, it exemplifies the trend of customizing LLMs for specific industries or use cases, trading generality for domain expertise [2].

- **Generative Pre-trained Transformer:**

  Generative Pre-trained Transformer (GPT) models, developed by OpenAI, are among the most widely used LLMs in natural language generation tasks. Built on the autoregressive transformer architecture, GPT models predict the next token in a sequence based on the preceding context. This design enables GPT to generate fluent, coherent, and contextually relevant text, making it ideal for content generation, document summarization, and conversational AI [6].

In summary, large language models and their multi-modal extensions are transforming document processing by enabling a deeper and more holistic understanding of content. They can read, summarize, and reason about documents at a level that approaches human-like comprehension in certain tasks. However, they also come with challenges: LLMs typically have limits on the amount of text they can handle at once, and they might require techniques like chunking or retrieval to deal with very large documents. Moreover, while they are powerful, ensuring the factual accuracy of LLM outputs is an active area of research, often addressed by incorporating knowledge graph checks, retrieval augmentation, or human-in-the-loop review as discussed above.

2.2.8 Knowledge Graphs

Knowledge Graphs represent information as networks of entities (nodes) and their relationships (edges). Figure 2.4 illustrates such a graph, where nodes like Product, Supplier, Customer, and Contract are interconnected through labeled relationships such as owns, supplies, or has terms. By linking related pieces of data in this way, a knowledge graph creates a structured, interlinked representation of otherwise fragmented information.

In document processing, this principle enables the organization and querying of information extracted from large collections of documents. For example, as shown in the figure, a knowledge graph could link a product to its supplier, supplier to its contracts, and contracts to their associated terms and countries. Similarly, in a medical setting, knowledge graphs could connect patient records, clinical notes, and laboratory results, revealing hidden relationships and improving data accessibility.

Beyond simple linkage, knowledge graphs support advanced applications such as semantic search (retrieving information based on meaning rather than exact keywords), recommendation systems (suggesting related entities or documents), and intelligent information lookup. They are especially powerful when combining data from many sources, both structured and unstructured, to provide a holistic view. In organizational settings, a knowledge graph might link internal documents, database records, and even external data like public datasets, enabling richer analytics and discovery of insights.

Recently, there has been growing interest in combining knowledge graphs with large language models (LLMs). LLMs can help populate and expand knowledge graphs by extracting structured facts from unstructured text, while knowledge graphs provide a factual backbone that helps LLMs reduce errors and hallucinations. This bidirectional synergy using LLMs to enrich graphs and graphs to ground LLM reasoning offers a promising path toward more reliable, interpretable, and verifiable AI-driven document analysis [7].

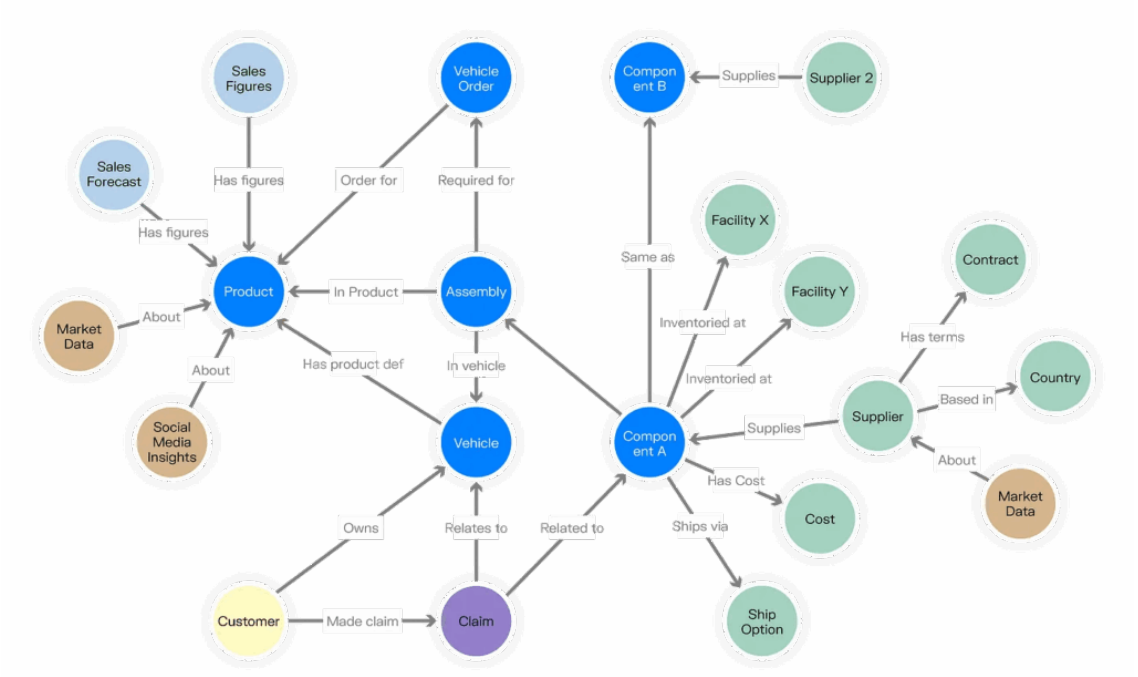


Figure 2.4: Example of a knowledge graph [3]

2.2.9 Multimodal Models

Multimodal models are neural architectures designed to process and integrate information from more than one modality—for instance, combining text, images, and layout structure within the same framework. In the context of document analysis, multimodal models jointly leverage visual features (e.g., page layout, typography), textual content (OCR or token embeddings), and sometimes structural cues (tables, form fields) to achieve a richer understanding of complex inputs. Unlike unimodal approaches, which treat documents as plain text or pure images, multimodal models can reason across modalities, making them especially effective for tasks such as information extraction, document classification, and layout-aware question answering.

A new class of vision–language models bypasses traditional OCR by processing document images end-to-end and directly producing structured outputs. One of the most prominent examples is the Document Understanding Transformer (Donut). As illustrated in Figure 2.5, Donut employs a vision encoder to transform the input page image into embeddings, which are then passed to a transformer-based decoder. Guided by task-specific prompt tokens, the decoder autoregressively generates a token sequence that can be parsed into structured formats such as JSON, containing fields like vendor, address, dates, totals, or line-items [27].

Other multimodal models have also shown strong performance in document processing. LayoutLMv3 extends the LayoutLM family by unifying text, image, and layout representations into a single transformer, achieving state-of-the-art results in tasks such as form understanding, receipt parsing, and document classification [24]. Similarly, DocFormer integrates visual features with textual embeddings through a hierarchical transformer, allowing fine-grained modeling of document layouts, tables, and complex structures [5]. Both models, like Donut, demonstrate the benefits of multimodal architectures in capturing the interplay between textual and visual information, which is critical for extracting accurate structured data from unstructured documents.

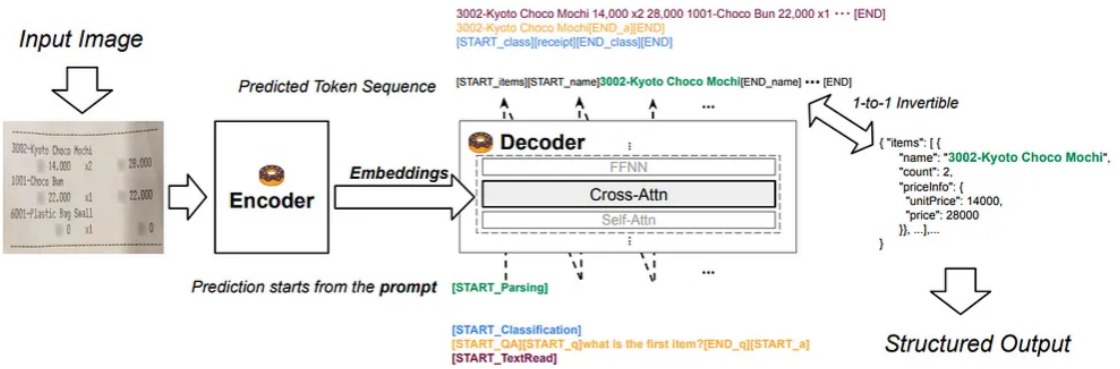


Figure 2.5: Donut Architecture

This architecture jointly models layout, typography, and semantics, enabling greater

robustness to irregular reading order, dense formatting, stamps, or noisy scans. By removing the OCR stage, it reduces latency and avoids error propagation between recognition and understanding. However, end-to-end models are often data-hungry, less modular than traditional pipelines, and prone to domain shift when facing unseen templates or languages. Successful deployment therefore requires careful fine-tuning on representative datasets (e.g. Portuguese invoices and receipts) and rigorous evaluation to ensure reliability.

2.3 Integration with Document Management Systems and Databases

After data is pulled from documents, the next step is ensuring that it is correctly integrated into a designated system usually a document management system (DMS) or a database where it can be stored, searched, and used by other applications. Effective integration ensures that the outputs of an AI document processing pipeline seamlessly feed into downstream systems [26]. The main strategies for integration today include:

- **Using Structured Data Formats:**

  To make the data easy to work with, it is usually converted into structured formats like JSON or XML. These formats help ensure that the information is organized in a way that other systems can understand and process smoothly [50].

- **Connecting Systems with APIs and Middleware:**

  APIs and middleware tools act as bridges between different software systems. They let the extracted data flow automatically into DMS platforms or databases without the need for manual copying and pasting. For example, a RESTful API can transmit the parsed document data from the AI extraction module directly into a records database or content management system. Middleware or integration platforms can also perform transformations and routing of data, ensuring compatibility between the AI system and legacy systems.

- **Validating and Cleaning the Data:**

  Before the data is sent off to other systems, it often goes through a cleanup and validation process [22]. This includes checking for errors or omissions, correcting inconsistencies, and making sure everything is formatted correctly (for instance, dates in the proper format, numeric fields within expected ranges, etc.). Good validation and cleaning help prevent problems later on and keep the integrated data accurate and reliable [49].

By using structured formats, robust APIs, and validation steps, organizations can ensure that the outputs of AI document processing are effectively integrated into business

workflows, triggering appropriate actions (like updating a database record, initiating a transaction, or storing the document in a repository) with minimal human intervention.

2.4 Summary

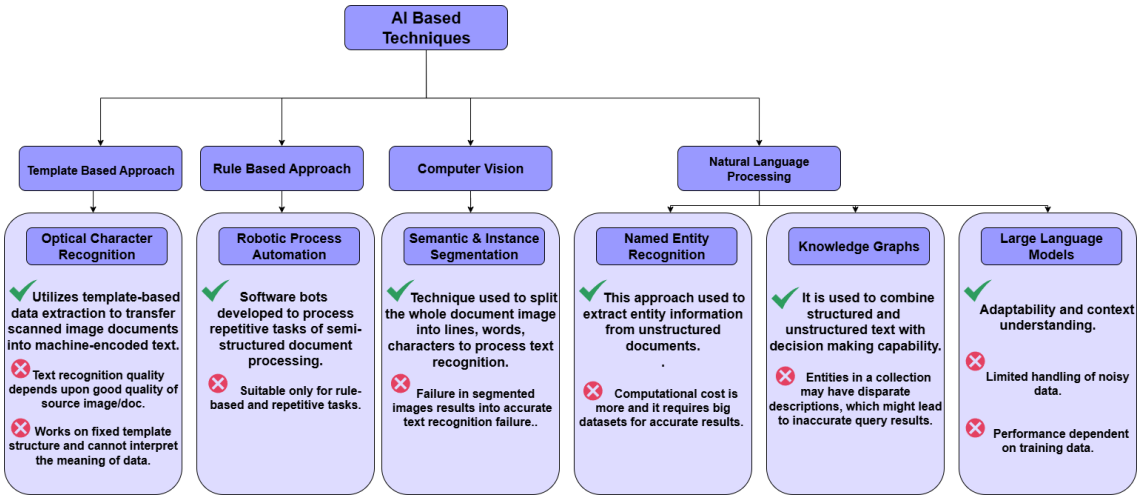


Figure 2.6: Unstructured document processing approaches using AI [31].

Figure 2.6 illustrates various AI-based approaches for unstructured document processing, highlighting their key functionalities and use cases. While each of these techniques plays a crucial role in automating document analysis, each approach has its own strengths and limitations. For instance, some methods are extremely fast but may handle only specific formats, whereas others are more general but computationally heavier.

To further understand the trade-offs between different methods, Table 2.1 provides a comparative analysis of several commonly used document processing techniques. The table evaluates each approach based on its advantages and disadvantages, offering insights into their suitability for different document types and processing needs. For example, OCR is excellent for transcribing text but cannot by itself understand document semantics; RPA is great for automating well-defined tasks but struggles with unstructured inputs; NER can pull structured data from text but requires large training sets and computational resources; and LLMs can interpret and generate text flexibly but have limitations with extremely long documents or complex 2D layouts.

This comparison helps in identifying the best-suited techniques for specific document processing challenges, particularly when handling large volumes of unstructured data. Often, a hybrid approach is warranted combining multiple techniques (like using OCR plus an LLM, or using semantic segmentation plus NER) to tackle the problem from different angles. By analyzing the characteristics of each method, researchers and practitioners can make informed decisions when selecting the most effective AI-driven approach for a given document understanding task. Ultimately, the state-of-the-art suggests that an integrated pipeline one that uses the right tool for each subtask and leverages the latest

AI advances stands the best chance of meeting the demands of accuracy, efficiency, and scalability in document processing.

Technique	Advantages	Disadvantages
Optical Character Recognition (OCR)	Efficient data input results in fewer data recognition errors. It converts scanned document data into easily editable forms, such as text and word.	Struggles with low-quality or noisy images, leading to errors. Formatting, layout, and non-text elements are often lost.
Robotic Process Automation (RPA)	Automates repetitive software tasks, saving time and reducing operational costs. Consistent execution of rule-based processes.	Performs poorly with unstructured or highly variable input. Lacks adaptability without explicit rules.
Semantic Segmentation	Distinguishes objects with pixel-level precision. No prior rules or templates needed. Useful for categorizing diverse visual elements.	Segmentation results can vary between annotators, lowering consistency. Requires high-level domain knowledge for meaningful categories.
Named Entity Recognition (NER)	E xtracts entities directly from unstructured text. Flexible, does not require pre-defined templates. Scales efficiently with well-annotated data. NER is efficient for automatically extracting entity information from unstructured documents.	For mixed text data recognition to produce better results, large datasets are required. Struggles with ambiguous or context-dependent entities.
Knowledge Graphs	Integrates structured and unstructured data. Enables reasoning and improved decision-making through explicit relationships.	Entity linking errors can propagate through the graph, reducing reliability. Scalability challenges arise with rapidly changing data.
Large Language Models	Handle diverse NLP tasks such as summarization, translation, and question answering. Capture semantic and contextual information in text.	They are limited to handling linear text and cannot inherently process two-dimensional information like tables or complex document layouts. LLM's have constraints on the amount of text they can process at once, necessitating strategies to segment extensive documents into manageable parts.
Multimodal Models	Bridge text, image, and other modalities, enabling richer understanding. Reduce error propagation between sequential OCR/NLP steps. More robust to noisy inputs.	Require very large, diverse training datasets.

Table 2.1: Comparison of data processing techniques

3

Dataset

3.1 Description

A crucial component of this research is the availability of annotated datasets that enable training, fine-tuning, and evaluation of the proposed approaches. Two distinct types of data were used, reflecting the dual nature of the system: semi-structured government contracts and visually complex invoices/receipts. This chapter describes the datasets, the preprocessing steps applied, and the rationale for their use.

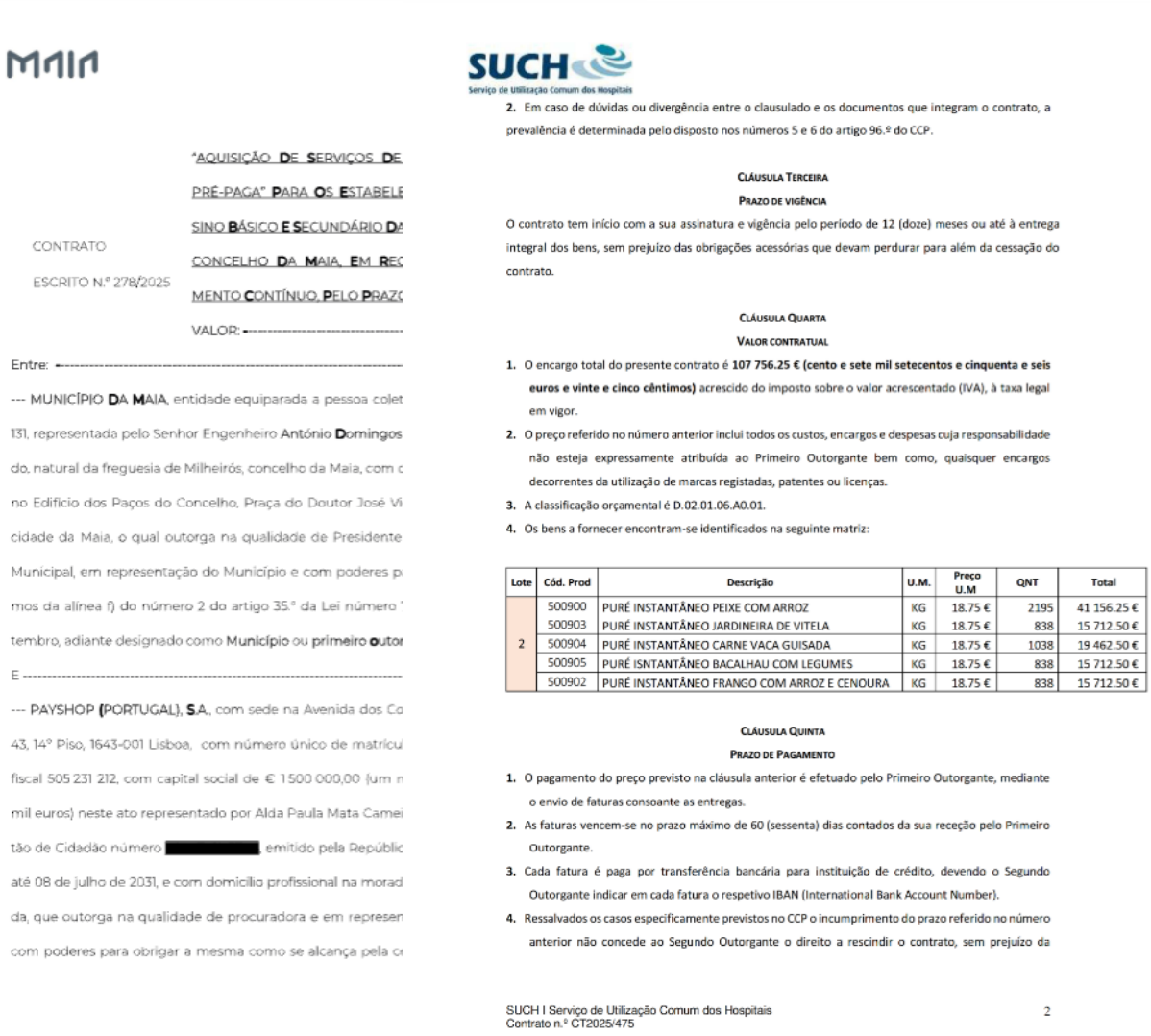
3.1.1 Government Contracts

The first dataset consists of a corpus of contracts collected from the official Portuguese government procurement portal, Base.gov.pt [36]. Each contract was annotated with key fields such as contractor name, date, address, and total contract value. On average, these contracts contained five pages per document, reflecting the length and level of detail typically present in such legal and administrative texts.

The dataset was created by programmatically scraping and downloading contracts from the Base.gov.pt portal, followed by a cleaning step to remove duplicates and incomplete files. Annotators then labeled each document with the relevant fields, producing a consistent, structured dataset suitable for downstream training and evaluation.

These documents represent semi-structured, domain-specific texts with the following characteristics:

- **Complex layouts:** Information is often distributed across multiple sections or pages.
- **OCR noise:** Scanned copies introduce recognition errors that must be handled.
- **Heterogeneous formatting:** Variability in templates and styles complicates extraction.



MAIA

"AQUISIÇÃO DE SERVIÇOS DE
PRÉ-PAGA" PARA OS ESTABELE
SINO BÁSICO E SECUNDÁRIO DA
CONTRATO CONCELHO DA MAIA, EM REG
MENTO CONTÍNUO, PELO PRAZ
ESCRITO N.º 278/2025
VALOR: ----------

Entre: ----------

--- MUNICÍPIO DA MAIA, entidade equiparada a pessoa colet
131, representada pelo Senhor Engenheiro António Domingos
do, natural da freguesia de Milheirós, concelho da Maia, com
no Edifício dos Paços do Concelho, Praça do Doutor José Vi
cidade da Maia, o qual outorga na qualidade de Presidente
Municipal, em representação do Município e com poderes p
mos da alínea f) do número 2 do artigo 35.º da Lei número
tembro, adiante designado como Município ou primeiro outo

E ----------

--- PAYSHOP (PORTUGAL), S.A., com sede na Avenida dos Co
43, 14º Piso, 1643-001 Lisboa, com número único de matrícu
fiscal 505 231 212, com capital social de € 1 500 000,00 (um
mil euros) neste ato representado por Alda Paula Mata Camei
tão de Cidadão número [redacted], emitido pela Repúblic
até 08 de julho de 2031, e com domicílio profissional na morad
da, que outorga na qualidade de procuradora e em represen
com poderes para obrigar a mesma como se alcança pela c

SUCH
Serviço de Utilização Comum dos Hospitais

2. Em caso de dúvidas ou divergência entre o clausulado e os documentos que integram o contrato, a prevalência é determinada pelo disposto nos números 5 e 6 do artigo 96.º do CCP.

CLÁUSULA TERCEIRA
PRAZO DE VIGÊNCIA

O contrato tem início com a sua assinatura e vigência pelo período de 12 (doze) meses ou até à entrega integral dos bens, sem prejuízo das obrigações acessórias que devam perdurar para além da cessação do contrato.

CLÁUSULA QUARTA
VALOR CONTRATUAL

1. O encargo total do presente contrato é **107 756.25 € (cento e sete mil setecentos e cinquenta e seis euros e vinte e cinco cêntimos)** acrescido do imposto sobre o valor acrescentado (IVA), à taxa legal em vigor.
2. O preço referido no número anterior inclui todos os custos, encargos e despesas cuja responsabilidade não esteja expressamente atribuída ao Primeiro Outorgante bem como, quaisquer encargos decorrentes da utilização de marcas registadas, patentes ou licenças.
3. A classificação orçamental é D.02.01.06.A0.01.
4. Os bens a fornecer encontram-se identificados na seguinte matriz:

Lote	Cód. Prod	Descrição	U.M.	Preço U.M	QNT	Total
2	500900	PURÉ INSTANTÂNEO PEIXE COM ARROZ	KG	18.75 €	2195	41 156.25 €
	500903	PURÉ INSTANTÂNEO JARDINEIRA DE VITELA	KG	18.75 €	838	15 712.50 €
	500904	PURÉ INSTANTÂNEO CARNE VACA GUISADA	KG	18.75 €	1038	19 462.50 €
	500905	PURÉ ISNTANTÂNEO BACALHAU COM LEGUMES	KG	18.75 €	838	15 712.50 €
	500902	PURÉ INSTANTÂNEO FRANGO COM ARROZ E CENOURA	KG	18.75 €	838	15 712.50 €

CLÁUSULA QUINTA
PRAZO DE PAGAMENTO

1. O pagamento do preço previsto na cláusula anterior é efetuado pelo Primeiro Outorgante, mediante o envio de faturas consoante as entregas.
2. As faturas vencem-se no prazo máximo de 60 (sessenta) dias contados da sua receção pelo Primeiro Outorgante.
3. Cada fatura é paga por transferência bancária para instituição de crédito, devendo o Segundo Outorgante indicar em cada fatura o respetivo IBAN (International Bank Account Number).
4. Ressalvados os casos especificamente previstos no CCP o incumprimento do prazo referido no número anterior não concede ao Segundo Outorgante o direito a rescindir o contrato, sem prejuízo da

SUCH I Serviço de Utilização Comum dos Hospitais
Contrato n.º CT2025/475 2

Figure 3.1: Examples of government contracts

3.1.2 Receipts and Invoices

The second dataset merges a collection of 2,003 annotated images derived from the InvoicesReceiptsPT [10] and the ICDAR 2019 SROIE [23] datasets. Both datasets contain receipts and invoices labeled with key fields such as company, date, address, and total amount. To ensure consistency, a schema alignment procedure was applied to harmonize label definitions across sources, and all data fields were standardized to a common format.

Invoices and receipts differ significantly from government contracts: they feature high visual complexity, diverse layouts, and strong reliance on spatial cues. These properties make them well-suited for evaluation of multimodal models such as Donut, which processes both visual and textual information jointly.

The use of two datasets with distinct characteristics enables a balanced evaluation: the OCR+LLM pipeline is naturally suited for text-heavy, semi-structured contracts, while

the multimodal Donut model is tested on visually complex receipts and invoices.



(a) Sample from InvoicesReceiptsPT



(b) Sample from ICDAR-2019 SROIE

Figure 3.2: Examples of invoices from the two datasets used in this thesis.

3.2 Preprocessing

Preprocessing ensured that documents were in a consistent, model-ready format. Different strategies were applied depending on the pipeline:

3.2.1 Government Contracts

For government contracts, the following preprocessing pipeline was applied:

- Attempt text extraction from embedded PDF layers using PDFMiner [39].
- When no text layer is available, page rendering at 300 DPI followed by OCR using `PyTesseract`.
- OCR was configured with Portuguese and English models (`lang="por+eng"`) to capture the bilingual nature of contracts.
- Extracted text is provided as raw unstructured input.

Field	Value
Contractor Name	GLUTAMINE - ALIMENTACÄO RACIONAL E DIETÉTICA. LDA.
Address	Rua do Outeiro 345 L-4485-029 Aveleda VCD
Date	30-07-2025
Total	107.756,25

Table 3.1: Relevant fields extracted from a government contract for the dataset.

3.2.2 Receipts and Invoices

For receipts and invoices, preprocessing focused on creating clean, standardized multimodal inputs:

- Images were normalized to RGB format and resized to 960×1280 pixels with padding.
- JSON annotations were converted into token sequences, with special tokens(e.g., <s_company>. . . </s_company>).
- The dataset was divided into 90% training (1,800 images) and 10% test (200 images)
- Dates were standardized, correcting variations such as "11 MAR 2022" or "2022-03-11" into a uniform representation.
- Total values were also normalized, ensuring consistent decimal notation (e.g., converting "10,5" into "10.5").

Token	Value
<s_company>	ADVANCO COMPANY
<s_date>	17/01/2018
<s_address>	JALAN WANGSA DELIMA 12, WANGSA LINK, WANGSA MAJU, 53300 KUALA LUMPUR
<s_total>	7.00

Table 3.2: Example of the annotated format for a sample invoice.

4

Proposed Solution

This chapter presents the architecture and design of the proposed system for automated data extraction from unstructured documents. The goal is to evaluate two complementary approaches: (1) a pipeline combining a traditional OCR engine (Tesseract) with a large language model (LLaMA 3.1) for field extraction, and (2) an end-to-end transformer-based model (Donut) trained directly on invoice and receipt images. By contrasting these pipelines, we aim to identify trade-offs between modular, interpretable workflows and unified, OCR-free architectures.

4.1 System Architecture

The system is designed to be modular and integrable with existing Document Management Systems (DMS). Figures 4.1 and 4.2 present the architecture diagrams of the two approaches under study: the Donut pipeline (Figure 4.1) and the OCR+LLM pipeline (Figure 4.2).

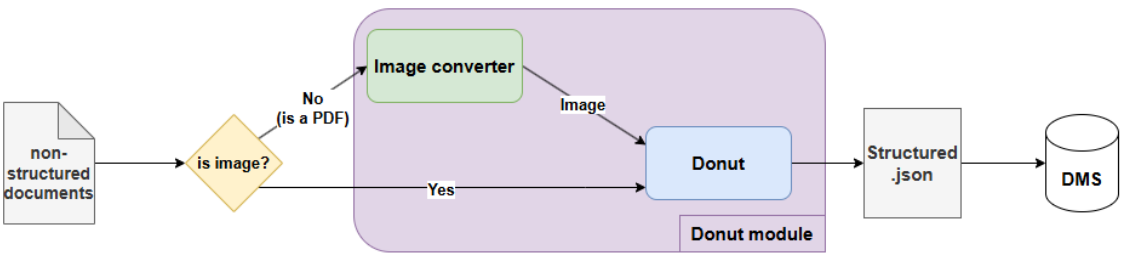


Figure 4.1: Pipeline Architecture diagram of the Donut approach

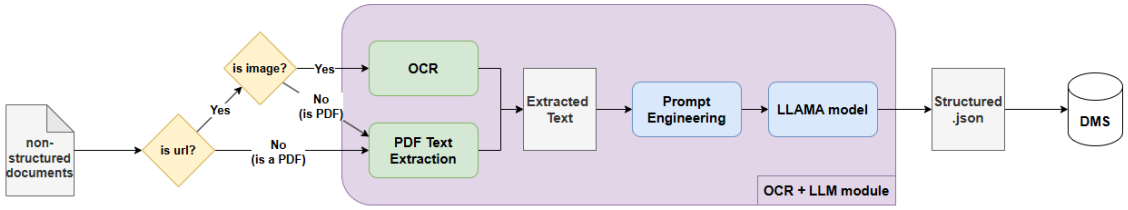


Figure 4.2: Pipeline Architecture diagram of the OCR+LLM approach

At a high level, the pipeline receives an incoming document of for example an invoice, a contract, or a receipt. It then processes it to extract structured information, and produces both machine-readable outputs, JSON and XML.

The architecture consists of the following components:

- **Document ingestion and pre-processing:**

  Two distinct strategies are followed to prepare documents for downstream extraction:

  1. **OCR + LLM pipeline:**

     Whenever possible, embedded text layers are directly extracted from PDF's using PDFMiner to avoid unnecessary OCR. If the document is scanned or lacks a text layer, Tesseract OCR is applied to generate a textual representation.

  2. **End-to-end Donut pipeline:**

     Documents are converted directly into image tensors. Images are normalized to RGB formart, and resized and padded to fixed dimensions (960×1280) to ensure compatibility with the Donut model.

- **Text extraction:**

  Two extraction strategies are implemented:

  1. **OCR + LLM pipeline:**

     The document is processed through Tesseract OCR if no text layer exists (for scanned images), or through PDFMiner when available. This produces raw text. Next, a local LLaMA 3.1 (8B) model performs structured field extraction, guided by prompt engineering to enforce schema-specific constraints (e.g., <company>, <date>, <total>, <address>). The use of carefully designed prompts ensures that the model consistently maps free-form OCR text into the desired output fields.

  2. **End-to-end Donut pipeline:**

     Instead of OCR, the Donut model by Kim et al [27], directly translates document images into structured outputs. Donut employs a Swin Transformer encoder for visual features and a BART-style decoder for text generation, enabling OCR-free processing that reduces error propagation. Prior studies show Donut achieves state-of-the-art performance on receipts and invoice extraction [27].

- **Data Integration:**

  The structured outputs generated by the extraction pipelines are delivered to the organization's document management system. This integration ensures that extracted fields become immediately available for downstream processes such as archiving, compliance monitoring, or automated reporting.

The system is designed to be modular and scalable, allowing for easy integration with existing workflows and adaptation to new document types or business requirements.

4.2 Technology Stack

The implementation of the proposed solution relies on a set of modern, open-source technologies that ensure reproducibility, scalability, and cost-effectiveness. The technology stack integrates three main layers: text recognition through an OCR engine, semantic field extraction with a large language model, and OCR-free document parsing using a transformer-based architecture. Together, these components create a flexible and efficient system capable of handling diverse document types.

4.2.1 Optical Character Recognition (OCR) with Tesseract

Optical Character Recognition (OCR) is a fundamental step in document processing pipelines, responsible for converting scanned or image-based inputs into machine-readable text. A wide range of tools exists in this domain, including both commercial and open-source solutions. While commercial services such as Google Cloud Vision, Microsoft Azure AI Vision [32] and Amazon Textract offer strong performance, they operate as subscription-based, black-box systems with limited transparency. In contrast, open-source alternatives prioritize accessibility and flexibility. Among these, Tesseract has emerged as the standard in open-source OCR, and it was selected as the OCR engine for this thesis.

- **Performance and Accuracy:**

  Tesseract has benefited from continuous development and community support since its adoption by Google. With the introduction of LSTM networks, its recognition accuracy improved substantially, particularly for clean, machine-printed documents. It also supports over 100 languages out-of-the-box, making it suitable for multilingual document processing.

- **Efficiency and Integration:**

  Compared to alternatives such as EasyOCR, OCRopus, or PaddleOCR, Tesseract provides a balanced trade-off between accuracy, efficiency, and ease of integration. While some newer tools may leverage deep learning for marginal improvements, they typically demand higher computational resources and more complex deployment.

Tesseract, by contrast, remains lightweight, portable, and easy to integrate into diverse environments.

- **Open-Source and Cost Advantages:**

  Unlike commercial OCR platforms, Tesseract is fully open-source, ensuring transparency, reproducibility, and long-term sustainability without recurring licensing costs. This makes it particularly suitable for research and enterprise deployments where cost control and system interpretability are essential.

In summary, Tesseract was chosen for this project because it combines robust accuracy with efficiency and accessibility, providing a practical and sustainable solution for the OCR+LLM pipeline.

4.2.2 Language Model for Field Extraction with LLaMA 3.1 (8B):

After text is obtained from the OCR stage, the next step is the extraction of structured fields, which requires a model capable of semantic understanding and generative output. For this task, LLaMA 3.1 (8B) was adopted, an addition to the LLaMA family of causal language models designed to deliver state-of-the-art performance with significantly fewer parameters than proprietary alternatives

- **Performance and Efficiency:**

  Despite its moderate size, LLaMA 3.1 (8B) benefits from architectural improvements that enable strong generative capabilities, achieving results competitive with much larger models. Meta's benchmark evaluations have shown that the smaller LLaMA 3.1 models approach the performance of larger proprietary systems, making them highly suitable for research and applied use. Crucially, the 8B variant strikes a balance between accuracy and computational feasibility: it requires only a single high-end GPU for inference, in contrast to the multi-GPU infrastructure demanded by models with hundreds of billions of parameters.

- **Adaptability and Context Handling:**

  LLaMA 3.1 introduces key features that expand its applicability to document processing. It supports a 128k token context window, enabling the model to process long-form documents without truncation—a significant advantage for contracts, invoices, and other extended inputs. In addition, it was trained on eight languages, allowing it to operate effectively in multilingual environments.

- **Deployment:**

  Running LLaMA 3.1 locally eliminates dependence on commercial APIs, ensuring reproducibility. Unlike closed-source services, it provides full transparency, allowing developers to inspect, adapt, and fine-tune the model as needed. This makes it

particularly well aligned with the goals of reproducible research and deployment in sensitive domains where external data transfer is undesirable.

4.2.3 OCR-Free Document Parsing with Donut:

Unlike traditional approaches, which first apply OCR and then pass the extracted text through a secondary parsing model, Donut integrates visual and textual understanding within a single transformer-based framework, enabling the model to interpret both layout and text jointly. This unified design eliminates error propagation between separate stages.

- **End-to-End Advantage:**

  By bypassing OCR entirely, Donut processes the document image holistically, learning to capture semantic meaning, formatting, and layout in a single pass. This reduces points of failure inherent in multi-stage pipelines, often resulting in more coherent and context-aware structured outputs. For tasks where document appearance plays a critical role (e.g., tables, multi-column layouts, or forms), Donut provides high performance.

- **Customizability and Fine-Tuning:**

  A major strength of Donut lies in its adaptability. While the base model offers strong generalization, performance can be significantly enhanced by fine-tuning on domain-specific datasets. This level of customization, combined with its open-source availability, provides flexibility beyond what proprietary, off-the-shelf solutions typically offer.

- **Comparison with Commercial Solutions:**

  Research has highlighted Donut's ability to rival, and in some cases surpass, commercial OCR-based services such as Amazon Textract or Microsoft Azure AI Vision. Whereas these services deliver high accuracy out-of-the-box, they lack the fine-grained adaptability of Donut and incur recurring licensing costs. Donut, by contrast, avoids subscription fees while enabling continual model improvement through iterative fine-tuning, offering both cost efficiency and long-term control over system performanc

4.2.4 Summary:

The technology stack highlights two complementary but contrasting approaches to document processing. The OCR+LLM pipeline, built on Tesseract and LLaMA 3.1, provides a modular and interpretable workflow where each step text recognition and semantic extraction can be monitored and optimized independently. This makes it well suited for scenarios involving text-native PDFs, multilingual inputs, or cases where transparency and debuggability are priorities. In contrast, the Donut approach adopts an end-to-end,

OCR-free architecture that directly generates structured outputs from document images. By jointly modeling layout and textual content, Donut often achieves higher accuracy on visually complex documents such as invoices and receipts, while avoiding error propagation from OCR.

In sum, the OCR+LLM pipeline emphasizes explainability, compatibility, and resource efficiency, whereas Donut prioritizes accuracy, adaptability, and streamlined processing. Comparing these two approaches enables a balanced assessment of trade-offs between transparency and performance in real-world document AI deployments.

4.3 Evaluation Measures

To ensure a fair comparison between the OCR+LLM and Donut approaches, a set of complementary evaluation measures was adopted. These metrics were designed to capture both token-level accuracy and field-level reliability, reflecting the practical requirements of document data extraction. For each metric, mean, median, and standard deviation values are reported across the evaluation set, enabling the identification of systematic errors (high mean error) versus instability (high variance) and providing a more nuanced understanding of overall model performance.

- **Field-Level Accuracy:**

  The primary measure of success is whether the system correctly extracts the target fields (<company>, <date>, <address>, <total>). A prediction is considered correct if the extracted value exactly matches the ground-truth annotation after normalization (e.g., standardized date formats, removal of diacritics).

- **Character Error Rate (CER):**

  CER quantifies similarity between predicted and reference strings at the character level. It is defined as:

$$CER = \frac{\text{Number of incorrect characters}}{\text{Total number of characters in the reference text}} \times 100\%$$

  This measure is particularly sensitive to small formatting errors (e.g., missing digits in totals or typographical mistakes in names).

- **Word Error Rate (WER):**

  WER evaluates predictions at the token (word) level, providing a higher-level perspective on extraction quality. It is defined as:

$$WER = \frac{\text{Number of incorrect words}}{\text{Total number of words in the reference text}} \times 100\%$$

This metric is especially relevant for fields such as company names or addresses, where whole-word accuracy is more important than minor character-level discrepancies.

- **Runtime Performance:**

  Since document-processing systems are typically deployed at scale, it is crucial to evaluate processing time per document. This measure captures whether a model can process large volumes of documents in near real-time.

- **Resource Consumption:**

  In addition to runtime, GPU/CPU utilization and memory usage are measured. This is particularly relevant when comparing the lightweight OCR+LLM pipeline to the more computationally intensive Donut model, as it reflects feasibility in resource-constrained deployment environments.

By integrating field-level accuracy with CER and WER, the evaluation captures both fine-grained precision at the character level and reliability at the word and field levels. The inclusion of runtime performance and resource consumption provides practical insights into the trade-offs between accuracy, efficiency, and scalability, ensuring that model performance is assessed not only in terms of correctness but also in terms of feasibility for large-scale, real-world deployments.

5

Methodology

This chapter describes the comparative experimental methodology used to evaluate automated information extraction pipelines for unstructured documents and invoices. Two distinct approaches were compared, an OCR + LLM pipeline and a end-to-end Donut transformer model.

Our aim is to identify which approach offers greater accuracy, robustness, and practical feasibility for real-world digital transformation in resource-constrained environments.

5.1 Model Architectures and Implementation

5.1.1 OCR + LLM Pipeline

The OCR + LLM pipeline follows a two-stage architecture, combining optical character recognition with schema-constrained large language model inference:

1. **Document Ingestion and OCR:**

   The pipeline can process documents from multiple sources:

   - Single PDF files provided via URL.
   - Local datasets.
   - Public datasets hosted on Hugging Face.

   For text extraction, two OCR backends are supported:

   - Tesseract (OCR engine).
   - PDFMiner (direct text extraction from PDF's)

2. **Field Extraction with LLM:**

   The recognized text is segmented into overlapping chunks (default: 7000 characters, with 400-character overlap) to respect context window limits. Each chunk is passed to a local LLaMA model (run via Ollama). The extraction step is controlled by:

   - A system prompt enforcing strict JSON-only responses and formatting rules.

- A user prompt specifying the schema to extract, which includes company name, date, address and total amount.

The model outputs a JSON object with all required fields, including placeholders for missing values.

This design makes the pipeline zero-shot, as the LLM is not fine-tuned on the task but constrained through prompt engineering and schema enforcement.

5.1.2 Donut Model

We fine-tune a Donut-style vision–language transformer instantiated via Hugging Face's VisionEncoderDecoderModel [19], using pre-trained weights from naver-clova-ix/donut-base [17]. The model is paired with a DonutProcessor [16] that handles both image normalization and target text tokenization.

- **Model initialization and tokenizer alignment:**

  We adjust the model so it can handle the vocabulary used for our task. This means expanding the decoder's word table (embedding matrix) to include any new special tokens from the tokenizer, ensuring they can be learned during training. The model is set to always begin generation with the <s> token, and padding tokens are marked so the model knows to ignore them.

- **Input sizing and sequence length:**

  To prevent encoder–preprocessor mismatches, the encoder's expected image size is synchronized with the processor's stored size. On the text side, the decoder max length is set to the maximum target length observed in the training labels to avoid truncation while keeping generation efficient.

- **Training setup:**

  Training is performed with Hugging Face's Seq2SeqTrainer [18] using mixed precision when the GPU is available. We train for 10 epochs with an effective batch size of 16 (per-device batch size 2 with gradient accumulation of 8) and weight decay 0.05 and the learning rate is set at 3×10^{-5}.

- **Objective:**

  The model is optimized using label-smoothed cross-entropy, with a smoothing factor of 0.1 applied over non-padding tokens. As shown in Figure 5.1, label smoothing distributes a small portion of the probability mass to incorrect classes instead of assigning full confidence to the correct class. This discourages overconfident predictions, improves generalization, and reduces susceptibility to overfitting.

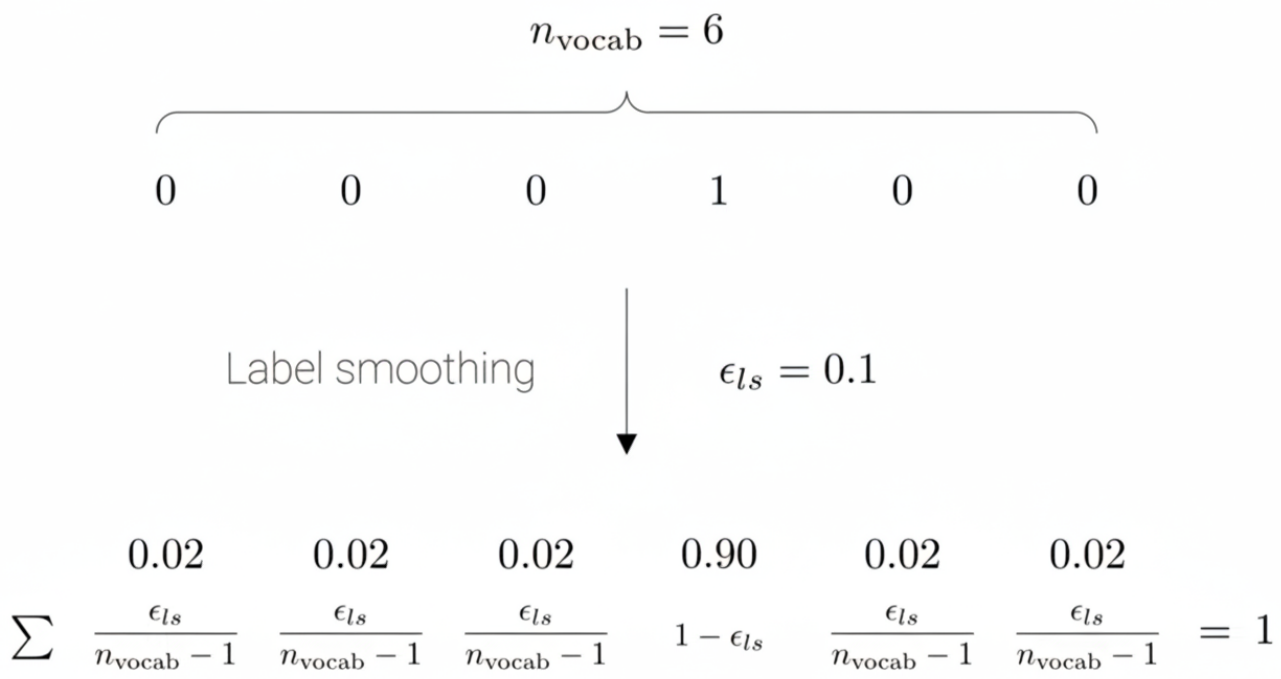


Figure 5.1: Example of label smoothing [11]

5.2 Environment

All experiments and evaluations were conducted on a cloud-based machine hosted on Amazon Web Services (AWS). Specifically, we used an EC2 instance of type g4dn.2xlarge, which provides:

- **GPU:** 1 × NVIDIA T4 GPU with 16 GB VRAM
- **vCPUs:** 8 Intel Cascade Lake vCPUs
- **Memory:** 32 GB RAM
- **Storage:** 200 GB NVMe SSD
- **Operating System:** Ubuntu 20.04 LTS

This instance type was selected as it offers a good balance between GPU capability and cost-effectiveness for deep learning experiments. The NVIDIA T4 GPU provides sufficient memory and performance to fine-tune medium-sized transformer models such as Donut-base, and to run inference with LLaMA 3.1 (8B) efficiently.

6

Evaluation

This chapter presents the evaluation of the proposed approaches for automated information extraction. The goal is not only to report performance metrics but also to interpret what they reveal about the strengths, weaknesses, and practical viability of the methods. This controlled test allows for robust statistical comparison.

The reported results are given as the mean and standard deviation over five runs. Repeating the experiments in this way reduces the impact of randomness in model inference and data shuffling, leading to more reliable statistics.

6.1 Experimental setup

- **Tasks and data:** Information extraction from invoices/receipts (single-page) and government contracts (multi-page).

- **Outputs:** For both approaches the output will be a structured JSON with fields: company, date, address, total

- **Measures:** Evaluation was carried out using Field-Level Accuracy, Character Error Rate (CER), Word Error Rate (WER), Runtime Performance, and Resource Consumption.

  - CER and WER were computed both on whole sequences and separately for each field.

  - Normalized variants of these metrics were also considered, where punctuation, spacing, and diacritics were ignored to focus on true semantic errors.

- **Robustness:** Each test is repeated 5 times on independent runs

6.2 Invoices and Receipts

6.2.1 Whole Sequence Errors

The first evaluation considers the predicted JSON output as a whole sequence. This provides an end-to-end view of performance.

Model	WER	CER
Donut	0.4068 ± 0.1425	0.1135 ± 0.0406
OCR + LLM	0.8561 ± 0.3010	0.3892 ± 0.1510

Table 6.1: Whole sequence errors (mean ± std) for invoices and receipts.

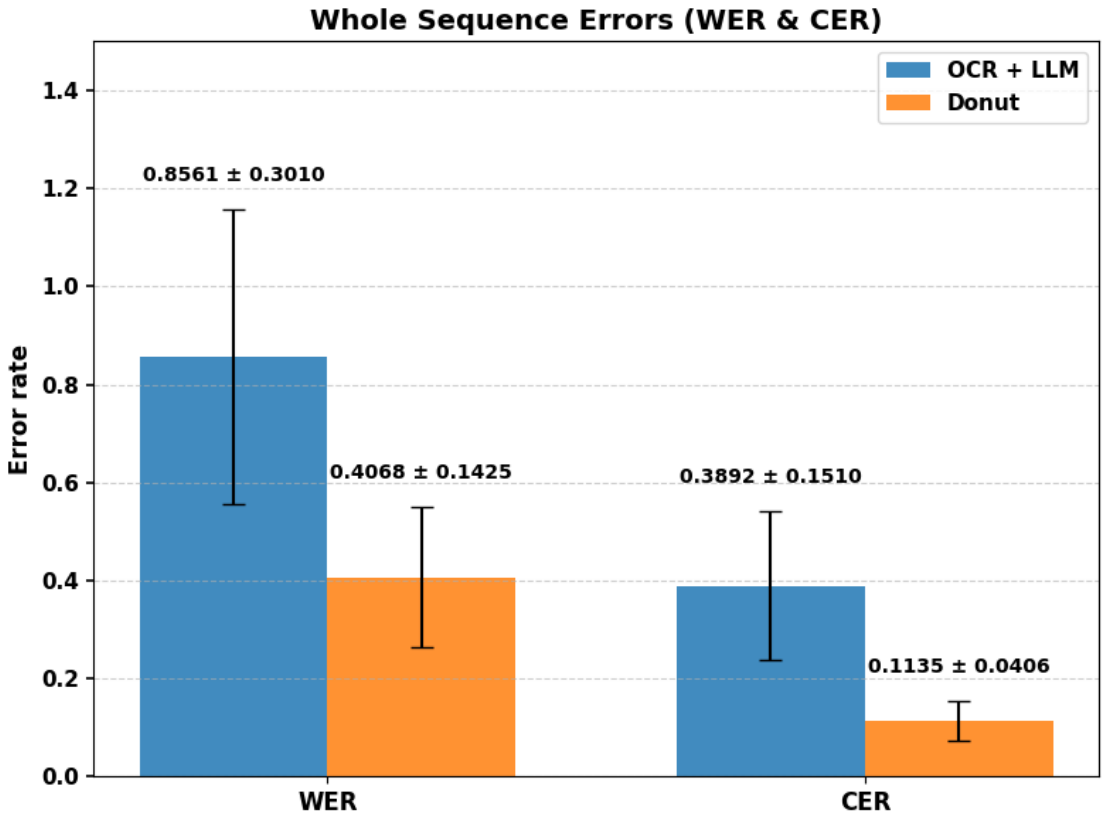


Figure 6.1: Whole sequence errors of both approaches

Table 6.1 and Figure 6.1 show that Donut clearly outperforms the OCR+LLM pipeline across both metrics. The difference is particularly noticeable in CER, which is highly sensitive to small transcription mistakes suggesting the possibility that OCR errors accumulate and propagate through the LLM, leading to a substantial drop in overall accuracy. Inspection of the OCR outputs confirmed this pattern: frequent character-level substitutions such as “0” vs. “O” or “1” vs. “l” were observed, which directly inflate CER values. Because the LLM never accesses the raw image, it cannot recover from such low-level errors once they enter the pipeline. The high standard deviation for OCR+LLM further

indicates instability: some documents were processed with comparatively low error rates, while others potentially collapsed due to noisy OCR outputs or misaligned parsing. By contrast, Donut demonstrated both higher accuracy and lower variance, reflecting more consistent performance across diverse invoice formats.

6.2.2 Field-Level Errors

To better understand which fields contribute most to the errors, we split the predictions and references by JSON field. We then compute WER and CER separately for each of the main fields: `<s_company>`, `<s_date>`, `<s_address>`, and `<s_total>`.

This allows us to analyze which fields are most error-prone and which are consistently reliable.

	Donut		OCR + LLM	
Field	**WER**	**CER**	**WER**	**CER**
Total	0.0752 ± 0.0166	0.0631 ± 0.0137	0.4982 ± 0.2594	0.3671 ± 0.2826
Date	0.1282 ± 0.0147	0.0619 ± 0.0049	0.4057 ± 0.2917	0.3253 ± 0.3062
Company	0.4010 ± 0.0809	0.2894 ± 0.0610	0.8014 ± 0.2346	0.7145 ± 0.2329
Address	0.2600 ± 0.0172	0.1338 ± 0.0106	0.7553 ± 0.2062	0.6840 ± 0.2781

Table 6.2: Field-level errors (mean ± std) on invoices/receipts.

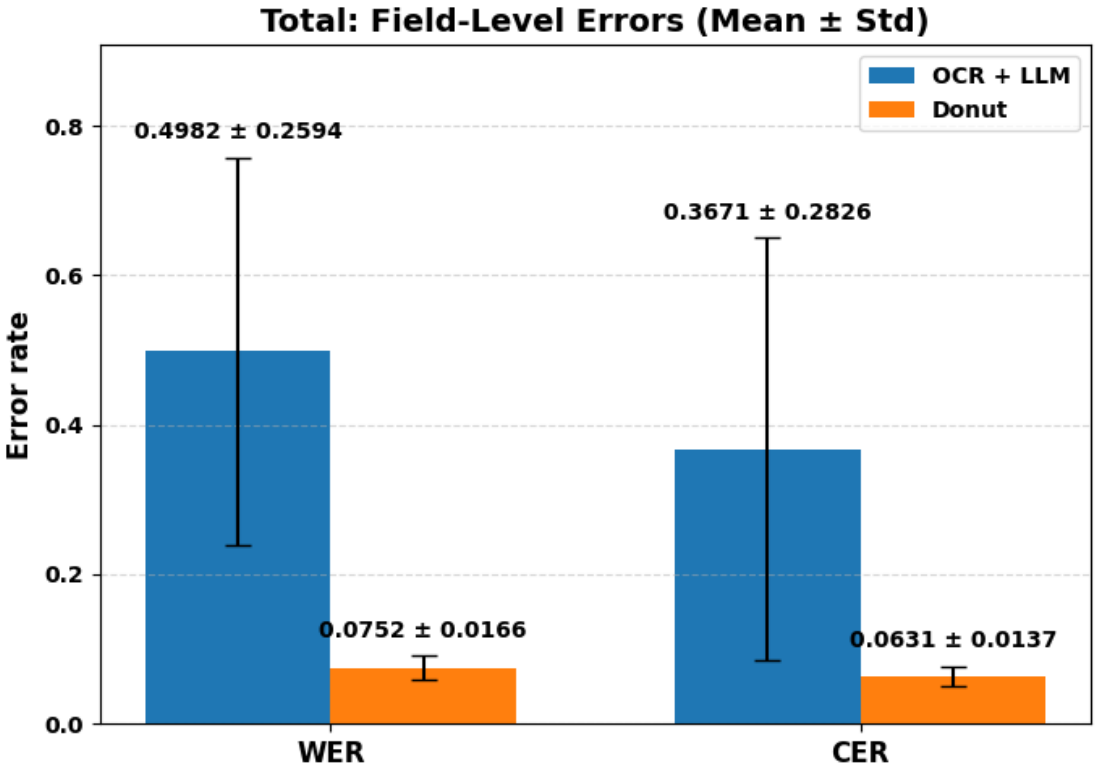


Figure 6.2: Field-level extraction errors for the Total field.

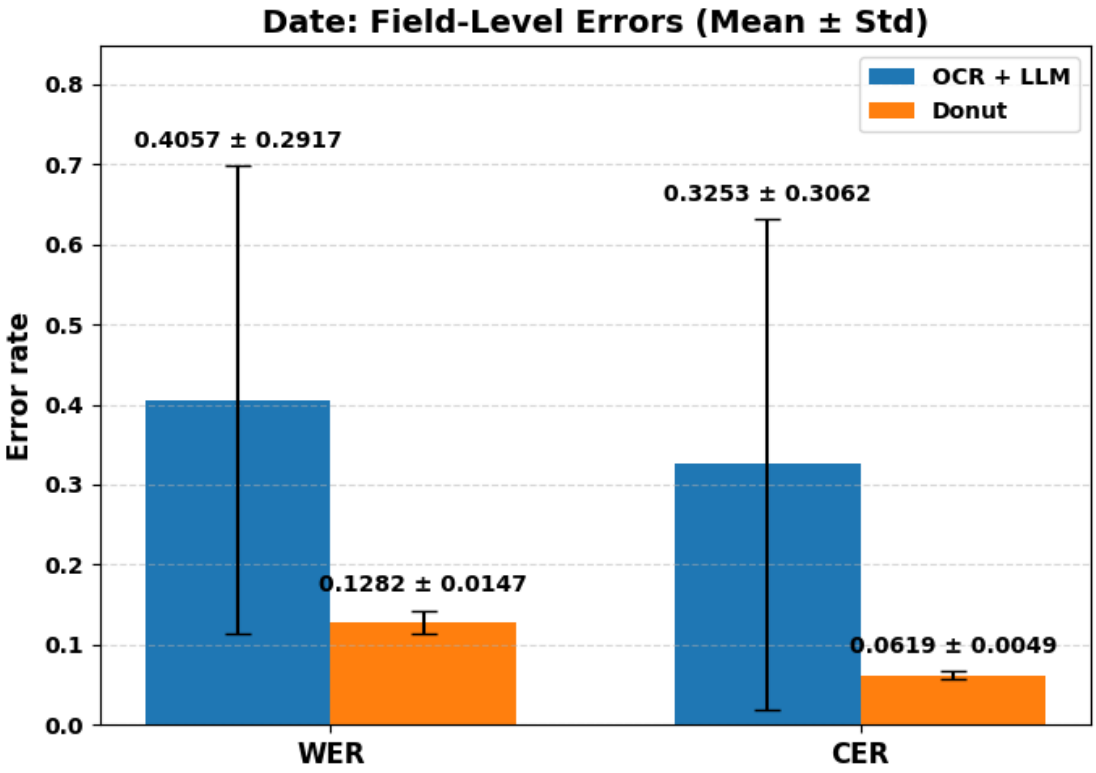


Figure 6.3: Field-level extraction errors for the Date field.

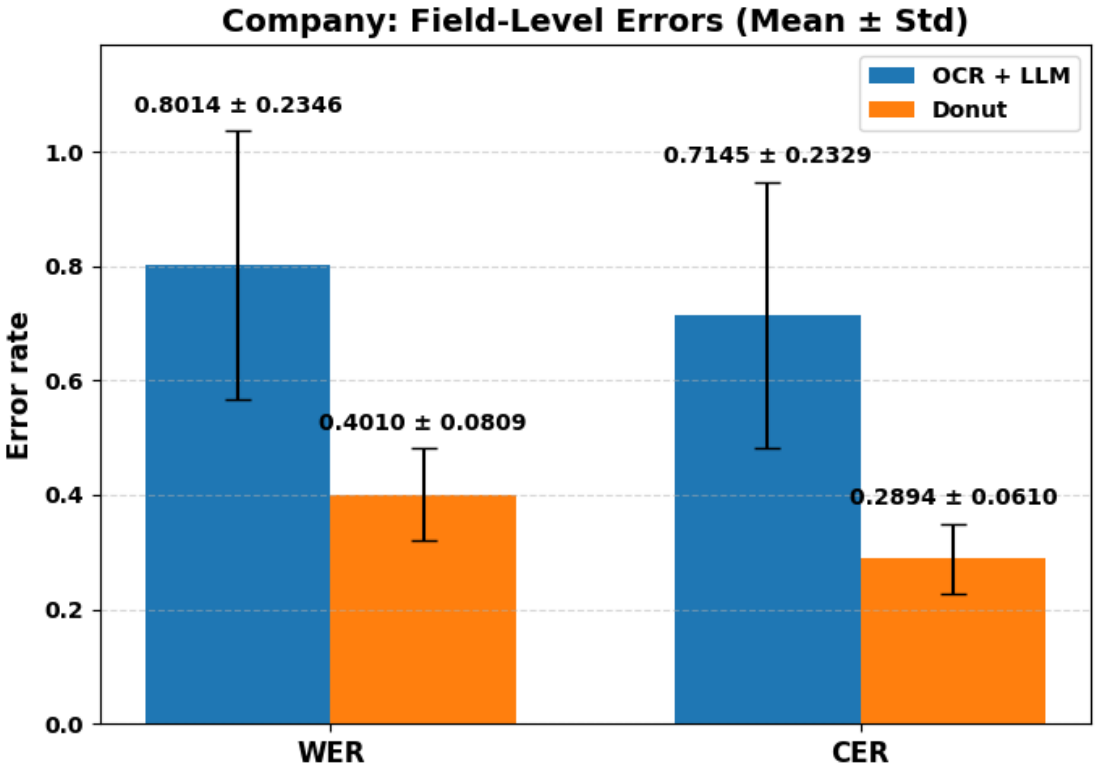


Figure 6.4: Field-level extraction errors for the Company field.

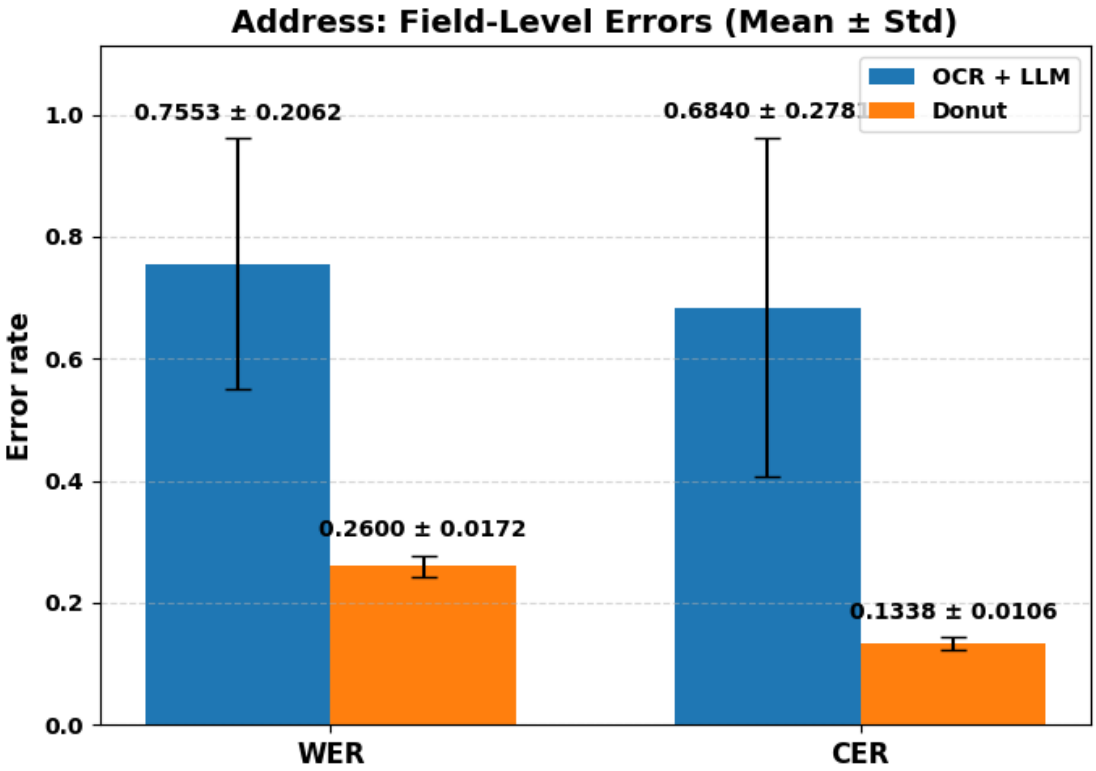


Figure 6.5: Field-level extraction errors for the Address field.

Table 6.2 and Figures 6.2-6.5 results reveal several important patterns:

- Totals are consistently the most accurate field, for both approaches. Numerical fields are less ambiguous and easier for models to capture.

- Dates also perform relatively well, but OCR+LLM often misinterprets small formatting variations ("11/03/22" vs. "11 MAR 2022"), whereas Donut is more robust. Because Donut operates directly on visual and textual cues without relying on a separate OCR transcription step, it learns multiple date patterns during fine-tuning. As a result, it is less sensitive to stylistic differences and better at generalizing across formats.

- Company names and addresses are the most error-prone. Invoices contain diverse vendor names, unusual typography, and diacritics (e.g., "João" or "calção"), all of which challenge OCR engines and subsequently the LLM. Donut, though clearly better, still struggled with long or noisy addresses.

- The much higher error rates of OCR+LLM suggest that once OCR introduces low-level mistakes in critical fields (e.g., confusing "0" with "O" or dropping diacritics), the LLM cannot reliably reconstruct the correct value because it does not have access to the original image.

From a practical standpoint, these findings are significant. While totals and dates may be sufficiently reliable for accounting tasks, high error rates in company names and addresses could disrupt business workflows.

6.2.3 Normalized Errors (Ignoring Formatting)

Normalization addresses minor formatting inconsistencies that can inflate error rates. Before computing WER and CER, all spaces, commas , and other formatting symbols are removed from both predictions and references.

This ensures that small formatting variations such as "10,50" vs. "10.50" or "Lisboa " vs. "Lisboa" are not counted as errors, focusing the evaluation on true content differences. Differences in the use of diacritics such as in "João" versus "Joao", or "calção" versus "calcao", would also inflate the CER and WER values. By normalizing such cases, the evaluation focuses on true semantic errors (e.g., wrong company name, incorrect numerical total, or mismatched date) rather than penalizing minor formatting or language-specific orthographic variations.

	Donut		OCR + LLM	
Field	**WER**	**CER**	**WER**	**CER**
Total	0.0495 ± 0.0157	0.0415 ± 0.0191	0.4882 ± 0.2301	0.3675 ± 0.2402
Date	0.1282 ± 0.0147	0.0619 ± 0.0049	0.4069 ± 0.2659	0.3251 ± 0.2502
Company	0.2845 ± 0.0141	0.2475 ± 0.0429	0.7236 ± 0.2152	0.6251 ± 0.2102
Address	0.4894 ± 0.0290	0.1226 ± 0.0112	0.7413 ± 0.1903	0.6278 ± 0.2404

Table 6.3: Normalized field-level errors (mean ± std) on invoices/receipts.

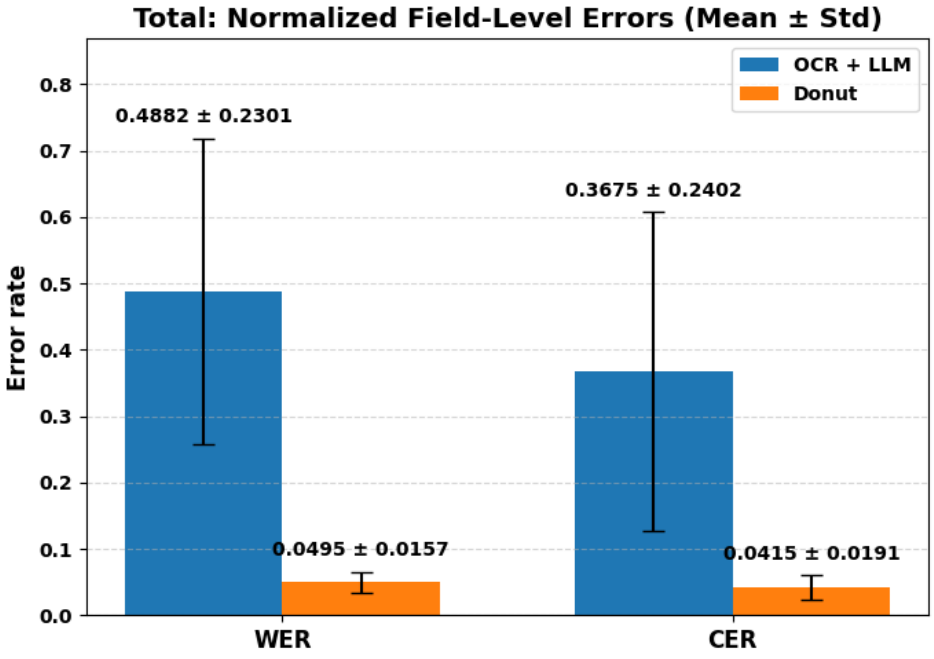


Figure 6.6: Normalized errors for the Total field.

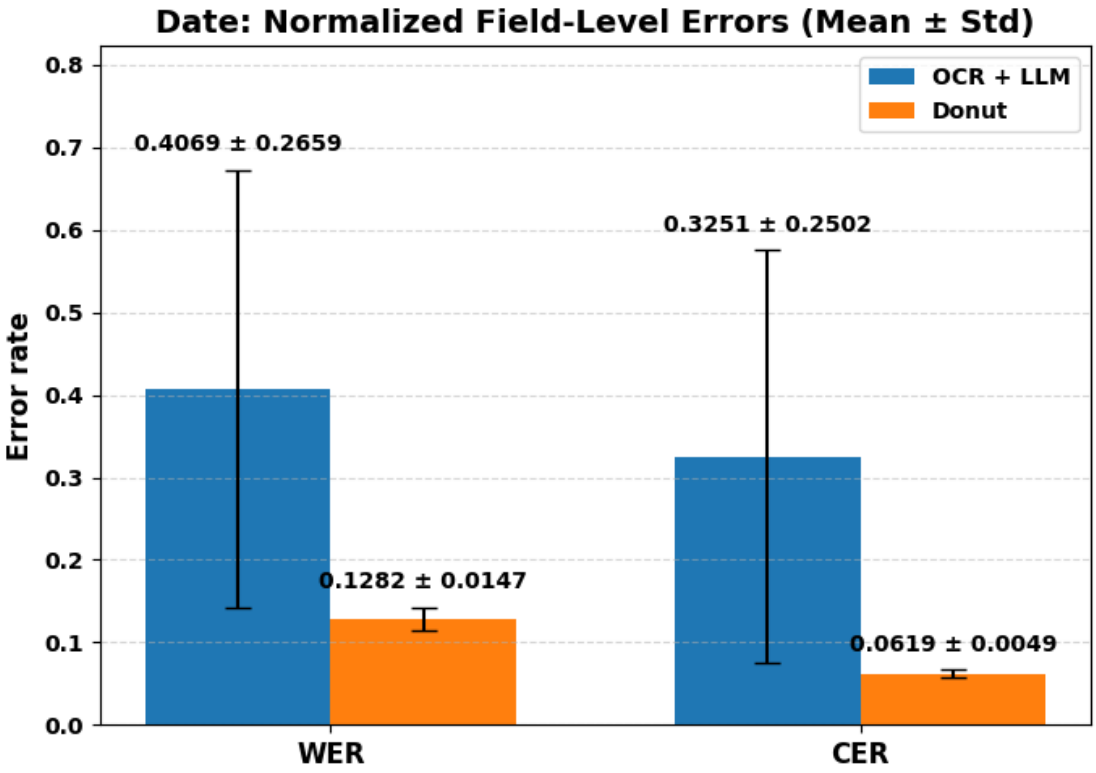


Figure 6.7: Normalized errors for the Date field.

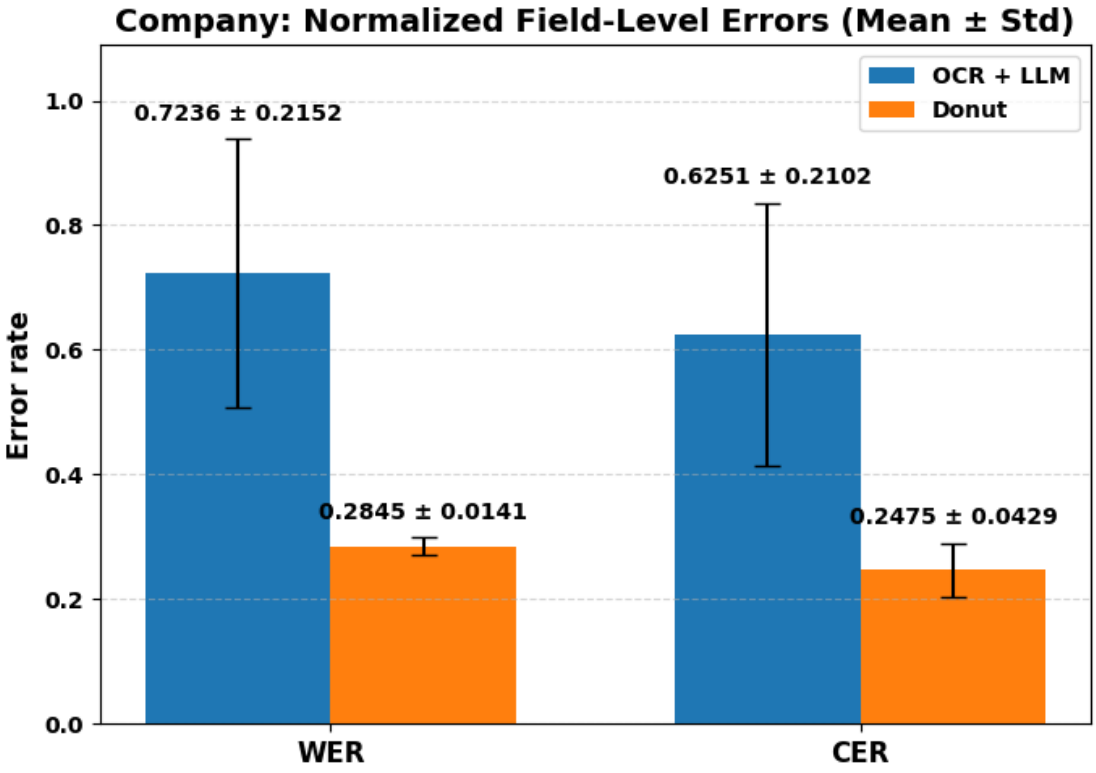


Figure 6.8: Normalized errors for the Company field.

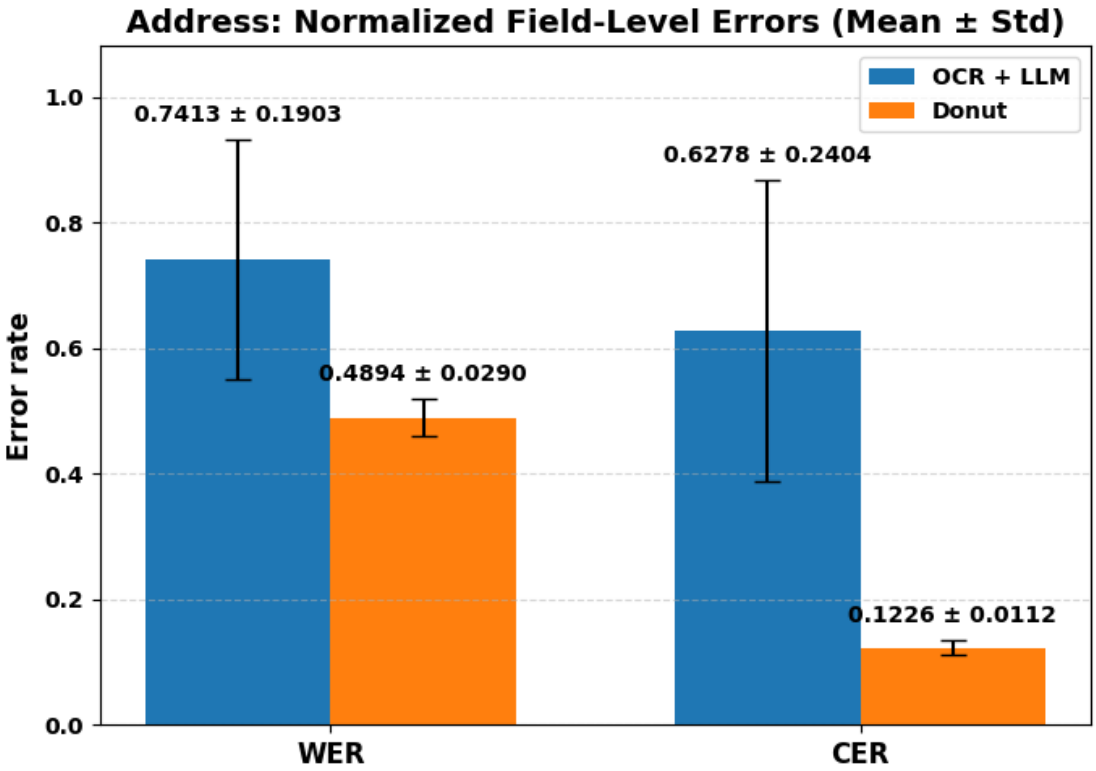


Figure 6.9: Normalized errors for the Address field.

Table 6.3 and Figures 6.6-6.9 show that normalized errors are slightly lower for both approaches, yet the relative advantage of Donut remained clear. This indicates that Donut's superiority cannot be attributed solely to handling superficial formatting inconsistencies (such as punctuation, spacing, or decimal separators). Instead, it reflects a more accurate understanding of the underlying document content, allowing it to recover field values even when layouts or formats vary.

The use of normalized evaluation is particularly important in multilingual contexts such as Portuguese. Orthographic conventions may introduce discrepancies that are orthographically visible but not semantically meaningful. The fact that Donut maintains its advantage under normalization therefore strengthens the claim that it captures the semantic structure of invoices more effectively than OCR+LLM.

6.2.4 Hardware Performance Comparison

The previous tests alone are insufficient for assessing practical viability; efficiency and scalability are equally critical. We therefore compared execution on CPU-only and GPU-enabled environments.

Approach / Setting	Execution Time	Avg. Time / Image
Donut (CPU-only)	1 h 58 min	35.2 s
Donut (GPU-enabled)	1 min 18 s	0.39 s
OCR+LLM (CPU-only)	1 h 54 min	33.75 s
OCR+LLM (GPU-enabled)	36 min 51 s	10.96 s

Table 6.4: Hardware execution times for processing 201 images, including average time per image.

Approach / Setting	RAM Usage	GPU Memory Usage
Donut (CPU-only)	6.0 GB	–
Donut (GPU-enabled)	1.2 GB	14.5 GB
OCR+LLM (CPU-only)	7.2 GB	–
OCR+LLM (GPU-enabled)	2.2 GB	5.3 GB

Table 6.5: Hardware memory usage for processing 201 images.

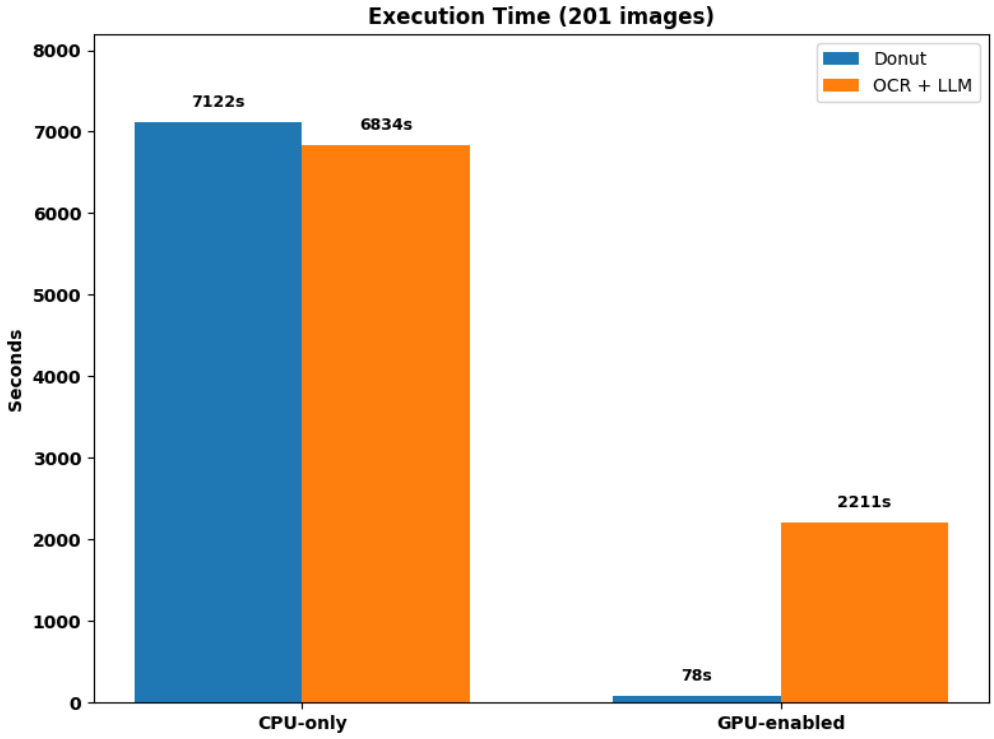


Figure 6.10: Total execution time of both approaches

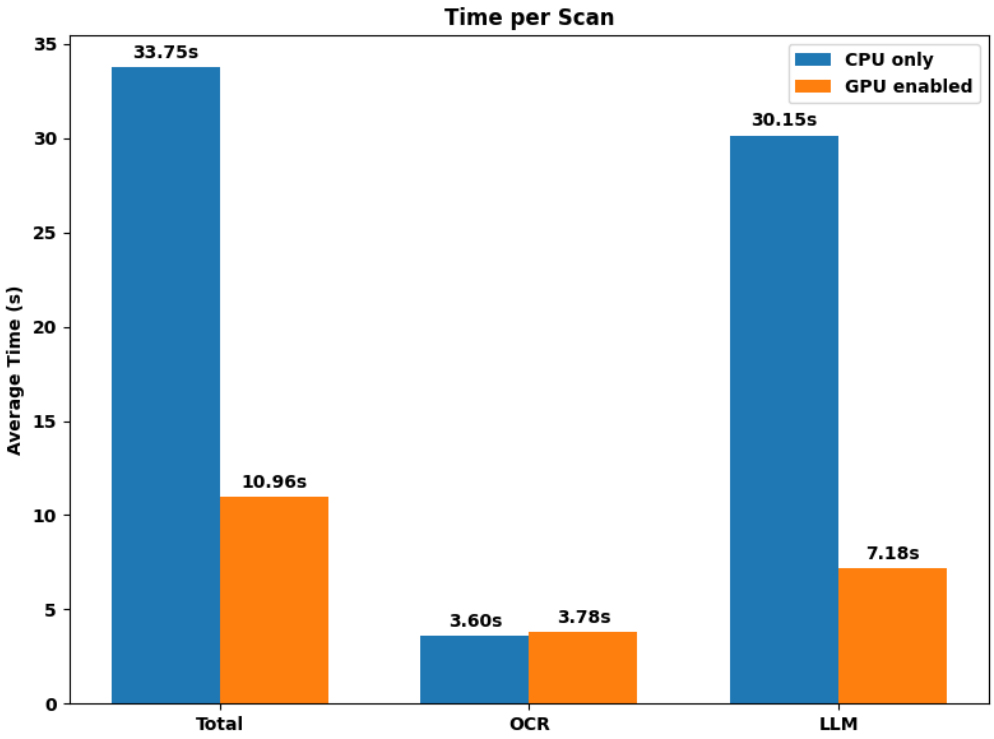


Figure 6.11: Scan Time breakdown of OCR+LLM

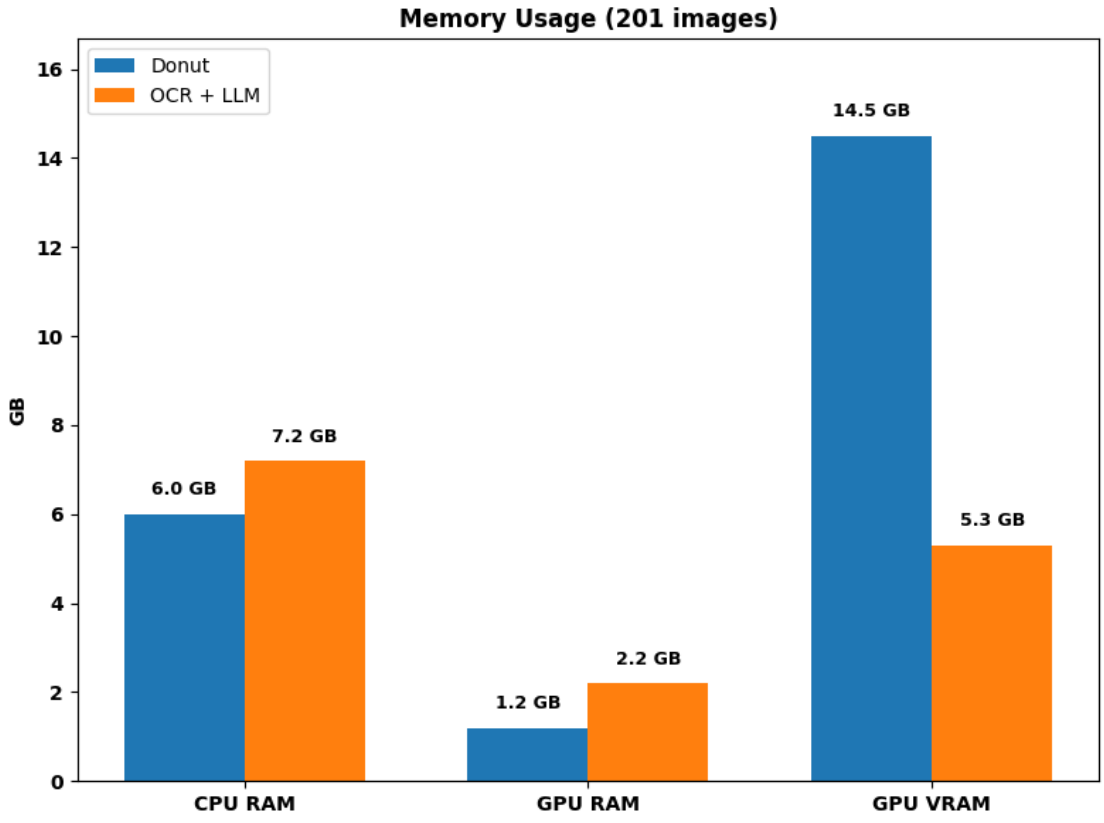


Figure 6.12: Memory usage of both approaches

Table 6.4 and Figures 6.10 and 6.11 highlight the dramatic improvements in runtime when leveraging GPU acceleration. Donut achieved nearly a 100× speed-up, processing the entire batch of 201 images in just over one minute. By comparison, OCR+LLM required more than half an hour even with GPU support, making it approximately 2,800% slower than Donut in this setting. This gap reflects the fact that OCR is heavily CPU-bound, limiting the benefits of GPU acceleration for the pipeline as a whole.

Resource trade-offs are also evident in Table 6.5 and Figure 6.12. Donut required substantially more GPU memory (14.5 GB), which may pose challenges for smaller organizations or resource-constrained public agencies. OCR+LLM, although significantly slower, was less demanding in terms of GPU VRAM (5.3 GB).

6.2.5 Summary of Evaluation

The evaluation provides several key insights:

1. Donut outperforms OCR+LLM consistently across all error metrics, confirming the advantages of end-to-end, OCR-free architectures for invoice-like documents.

2. Error distribution matters: while totals and dates are reliably extracted, company names and addresses remain challenging, reflecting the linguistic diversity and layout complexity of invoices.

3. Normalization highlights semantic accuracy: by removing orthographic noise, Donut demonstrates resilience in linguistic settings.

4. Hardware trade-offs are critical: Donut is efficient only with GPU acceleration, but its high VRAM demand raises deployment considerations. OCR+LLM, although slower, may remain relevant in contexts where GPUs are unavailable.

In summary, the results position Donut as the more accurate and scalable solution for invoice processing, while OCR+LLM remains a complementary option for contract documents where structured layouts differ. This duality underscores the importance of adapting document AI architectures to domain-specific requirements rather than seeking a single universal pipeline.

6.3 Government Contracts

Government contracts differ significantly from invoices and receipts. They are longer, multi-page PDF documents, often five pages or more, with semi-structured text-heavy layouts. This poses distinct challenges for automated extraction.

6.3.1 Donut Limitations on Contracts

The Donut model was tested on the contract dataset by sending each page of a PDF as an image input. However, the results were not satisfactory. While invoices and receipts fit Donut's design well, government contracts posed significant challenges.

Processing multi-page documents with Donut faces three critical obstacles:

- **Page-by-page restriction:** Donut cannot natively accept multi-page PDFs as input. Each page must be converted into a separate image and processed independently, which breaks the global context of the document.
- **Contextual continuity:** Contract fields (e.g., company details or total amounts) may appear across different pages. Donut lacks the ability to aggregate multi-page outputs into a unified structured record.
- **Scalability:** Fine-tuning Donut on contracts would require significant annotation effort, as each page would need to be aligned with JSON field targets.

6.3.2 Potential Solutions for Donut on Contracts

While the current Donut implementation is not directly applicable, several strategies could make it feasible in future work:

- **Page-wise Processing + Post-Processing:** Treat each page as a separate input to Donut, then merge page-level outputs using heuristic rules or a secondary model to reconstruct the contract-level JSON.
- **Hierarchical Models:** Extend Donut with a multi-page encoder that captures inter-page relationships. Recent research (e.g., LayoutLMv3 [24], DocFormer [5]) explores hierarchical encoders that could be adapted for contracts.
- **Hybrid Approach:** Use Donut to extract visual cues (e.g., headers, tables, signatures) while relying on OCR+LLM for text-heavy sections. A fusion mechanism could combine the outputs.

6.3.3 OCR + LLM

The OCR + LLM pipeline was successfully applied to this dataset. Using MinerU [45], text was extracted from the PDF text layer or, when unavailable, Tesseract[42] was used. The resulting text was then passed to the LLaMA model with a predefined schema.

6.3.3.1 Whole Sequence Errors

Model	WER (mean ± std)	CER (mean ± std)
OCR + LLM	0.9837 ± 0.3415	0.4624 ± 0.1827

Table 6.6: Whole sequence errors (mean ± std) using OCR+LLM.

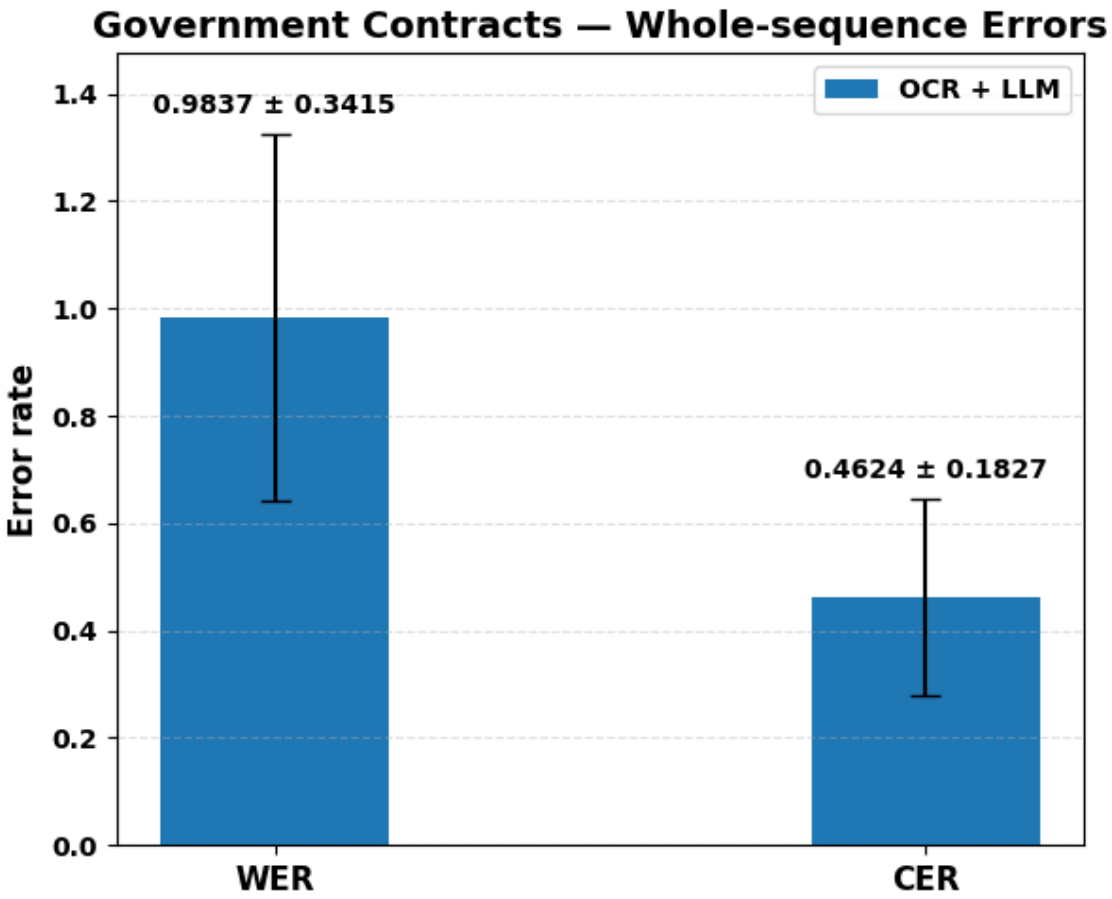


Figure 6.13: Whole sequence errors of OCR + LLM

Table 6.6 and Figure 6.13 show that the OCR+LLM pipeline produced high whole-sequence error rates, with WER close to 1.0 and CER around 0.46. These values are broadly consistent with those reported earlier for receipts, suggesting that the challenges are not unique to one document type. A possibility is that OCR mistakes such as misrecognized characters or missing accents enter the pipeline at an early stage and propagate through the LLM, and the high standard deviation also points to instability across documents.

6.3.3.2 Field-Level Errors

Field	WER	CER
Total	0.5632 ± 0.2974	0.3821 ± 0.2749
Date	0.6847 ± 0.4235	0.3438 ± 0.2426
Company	0.9823 ± 0.4027	0.7816 ± 0.2712
Address	0.8794 ± 0.2468	0.8629 ± 0.2591

Table 6.7: Field-level errors (mean ± std) using OCR+LLM.

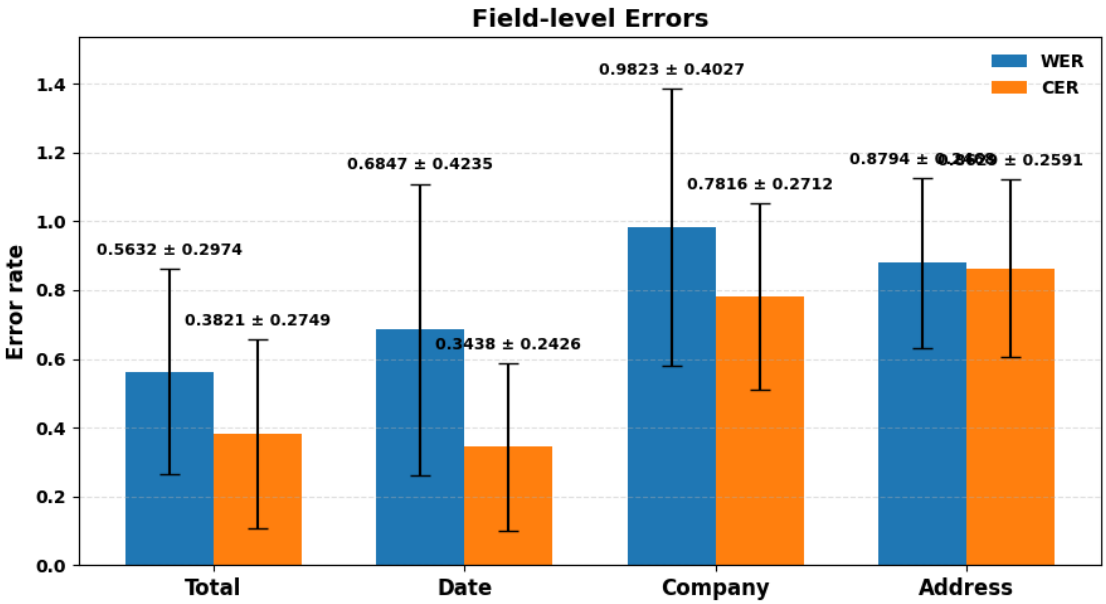


Figure 6.14: Field level errors of OCR + LLM

The field-level results in Table 6.7 and Figure 6.14 reinforce this trend. Totals and dates achieved lower error rates compared to company names and addresses, which remain particularly difficult to extract. This mirrors the pattern seen in receipts, where numerical and structured fields were relatively robust while long, variable text fields led to large error rates. One possible reason is that totals and dates have more predictable formats, making them less sensitive to OCR variability. By contrast, company names and addresses exhibit greater diversity, often involving uncommon words, diacritics, or multi-line layouts, which makes them especially vulnerable to OCR noise. Another contributing factor may be the absence of domain-specific fine-tuning, the LLM was applied with a general schema rather than being optimized for the linguistic and structural characteristics of contracts.

6.3.3.3 Normalized Errors

Field	WER	CER
Total	0.5445 ± 0.2722	0.3517 ± 0.2293
Date	0.6678 ± 0.3796	0.3242 ± 0.2184
Company	0.8849 ± 0.3624	0.6657 ± 0.2375
Address	0.8642 ± 0.2249	0.7753 ± 0.2317

Table 6.8: Normalized field-level errors (mean ± std) using OCR+LLM.

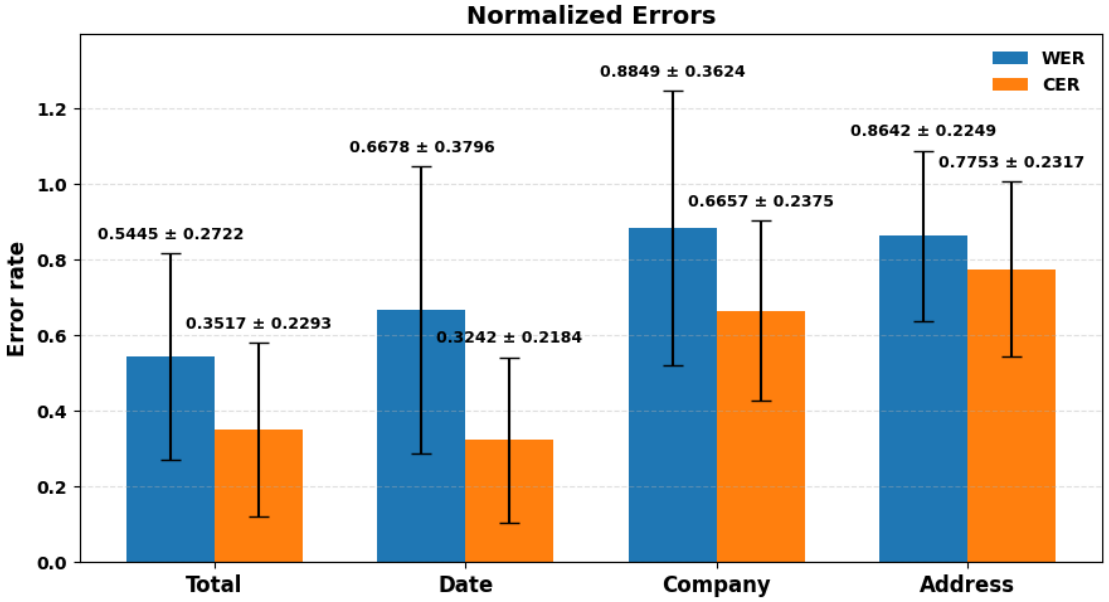


Figure 6.15: Normalized errors of OCR + LLM

Normalization reduced the error values slightly (Table 6.8 and Figure 6.15), but the overall trends remained the same. This indicates that formatting differences such as decimal separators or spacing are not the main drivers of error. Instead, the high WER and CER values point to deeper issues, such as missing words, incorrect company names, or truncated addresses. The similarity of the normalized results between contracts and receipts further supports the hypothesis that OCR mistakes are a central bottleneck for this pipeline. Without direct access to the visual structure of the page, the LLM cannot correct low-level transcription errors, limiting its effectiveness even when surface-level formatting inconsistencies are removed.

6.3.3.4 Hardware Performance

To better understand the runtime behavior of OCR+LLM under different hardware configurations, Table 6.9 and Figure 6.16 reports the average processing time per scan when executed on CPU-only and GPU-enabled environments.

Component	CPU only	GPU enabled
Total	113.72 s	33.72 s
OCR	23.29 s	19.19 s
LLM	90.43 s	14.52 s

Table 6.9: Average time per scan under CPU-only and GPU-enabled configurations.

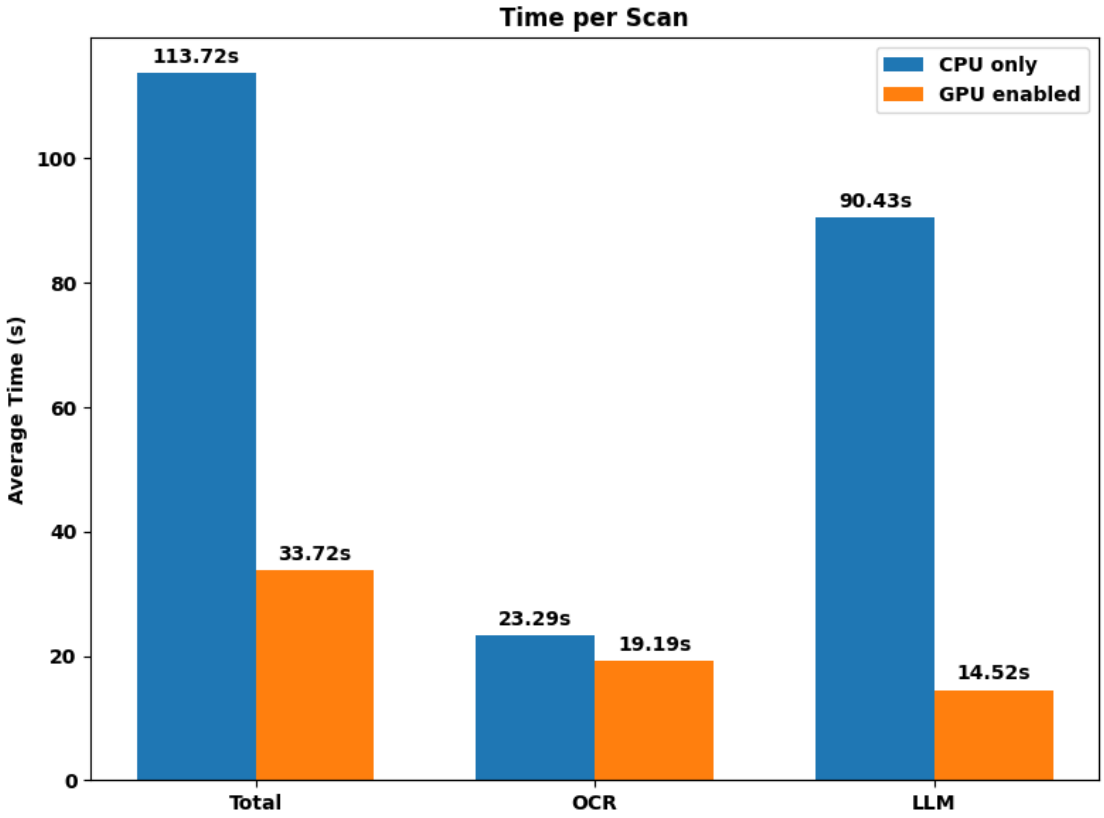


Figure 6.16: Scan Time breakdown of OCR+LLM

The results highlight that while GPU support reduces the total wall-clock time, the performance gains are modest, as the OCR stage remains largely CPU-bound, while the LLM component benefits more noticeably from GPU acceleration.

6.3.4 Evaluation

For government contracts, the OCR+LLM pipeline remains the more practical solution under current constraints, while Donut is not yet suitable for multi-page PDFs without substantial adaptation. Future work could explore hybrid or hierarchical strategies to extend Donut's applicability to long-form administrative documents.

The OCR+LLM approach proved effective for contracts because:

- The documents are rich in continuous text, rather than dense visual layouts.
- OCR performed reasonably well on these well-structured, machine-generated PDFs.

- The LLM could handle full-page text input, identifying relevant fields such as contractor, date, and total value.

The main limitation was OCR noise in scanned contracts (blurred or poorly digitized), which occasionally caused extraction failures. Nevertheless, the modular pipeline remains suitable for this type of content.

7

Discussion

7.1 Overview

This chapter interprets the experimental results presented in the previous chapter, moving beyond numerical metrics to analyze their implications for real-world document processing. The evaluation showed that neither approach is universally superior: Donut consistently outperformed OCR+LLM in invoices and receipts, while the OCR+LLM approach proved more suitable for government contracts. These outcomes reflect fundamental differences in how the two paradigms handle layout complexity, text density, and document length.

The discussion that follows examines these trade-offs in greater detail. It contrasts the strengths and weaknesses of each approach, highlights recurring error patterns, and situates the findings within the broader context of deployment feasibility. By doing so, it clarifies the practical implications of the results and identifies directions for improving intelligent document processing systems in both research and applied settings.

7.2 Comparative Analysis of Approaches

The comparative evaluation highlights clear differences between the two paradigms. Donut consistently achieved lower Character Error Rates (CER) and Word Error Rates (WER) across invoices and receipts, underscoring the strength of end-to-end multimodal architectures in handling visually complex documents. Its ability to jointly model layout, text, and semantics allowed it to overcome many of the limitations inherent to OCR-based pipelines, particularly in scenarios with noisy scans, irregular vendor layouts, or diacritic-rich company names. In contrast, the OCR+LLM pipeline suffered from error propagation: when OCR misrecognized a token (e.g., "João" becoming "Joa0"), the large language model was not always able to reconstruct the correct field, leading to higher error variance.

Despite these advantages, Donut's superiority is not universal. In the government contract dataset, OCR+LLM proved more robust. Contracts, unlike receipts, are predominantly text-heavy and extend across multiple pages. Here, the modular pipeline benefited

from Tesseract's ability to capture full-page text. Donut, by contrast, was constrained by its single-page input format, resulting in fragmented outputs that failed to preserve cross-page continuity. This divergence illustrates that document type strongly mediates the suitability of each approach: Donut excels in short, image-like documents, whereas OCR+LLM remains more effective for long, text-centric workflows.

Field-level analysis further revealed the trade-offs. Totals and dates were extracted with high reliability by both systems, suggesting that numerical fields are relatively resilient to recognition errors. However, addresses and company names consistently emerged as the most error-prone fields. Invoices contain stylized logos and unusual typographic conventions, which challenged Tesseract's OCR pipeline. While Donut performed better in these fields, its errors were more systematic for example, truncating long addresses or misclassifying accented characters. This indicates that while Donut reduces variance, it does not eliminate the underlying difficulty of extracting high accuracy text from real-world documents.

7.3 Practical Considerations

Beyond accuracy, the choice between Donut and OCR+LLM is shaped by practical deployment factors, particularly hardware requirements. Donut proved highly efficient on GPU, but achieving this performance required nearly 15 GB of dedicated VRAM, which may exceed the resources available to many environments operating on CPU-only servers or constrained cloud budgets. Even so, Donut still outperformed OCR+LLM in CPU-only settings, delivering better accuracy while using less RAM (6.0 GB vs. 7.2 GB).

Scalability further differentiates the two. With GPU acceleration, Donut processed invoices nearly a hundred times faster, making it a more viable option for high-volume workflows. Its resource demands, however, raise concerns in environments where GPU access is limited. OCR+LLM, while less demanding on GPU memory (5.3 GB), was consistently slower and less accurate, offering no clear advantage beyond lower hardware intensity for invoice type documents.

Integration flexibility also differs between the two approaches. OCR+LLM follows a modular design: individual components such as the OCR engine or the language model can be upgraded independently without retraining the entire system. Donut, while generally monolithic, can still be adapted with reduced cost by freezing large portions of the model and fine-tuning only the task-specific layers. This strategy mitigates some of the annotation and computational overhead, though it still requires more specialized expertise than simply swapping out components in a pipeline. As a result, Donut is well-suited for stable domains where minimal retraining suffices, while OCR+LLM retains an advantage in environments that demand frequent adjustments.

Together, these considerations underline that accuracy alone does not determine suitability. The selection of approach must reflect environment constraints, including available hardware, budget, and the diversity of document types encountered.

7.4 Error Patterns and Failure Cases

Error analysis revealed systematic weaknesses in both approaches that explain the aggregate performance differences. In the OCR+LLM pipeline, most failures originated in the OCR stage. Diacritics and special characters were frequently misrecognized (e.g., "João" becoming "Joa0"), which then propagated into incorrect extractions that the LLM could not recover. Similarly, noisy scans or blurred PDF layers led to partial token loss, producing incomplete company names or truncated addresses. These errors introduced high variance, with some documents being processed correctly and others failing.

Donut, by contrast, showed fewer failures but exhibited more consistent error patterns. Long addresses were often truncated or simplified, reflecting the model's difficulty in balancing sequence length with structured output constraints. Vendor names containing rare characters or non-standard typography were occasionally misinterpreted, even when the overall layout was correctly parsed. Importantly, Donut struggled with multi-page continuity in contracts: company names on the first page and totals on the final page were not linked, resulting in incomplete or fragmented outputs.

Normalization reduced the apparent error rates by filtering out superficial differences in punctuation, spacing, or diacritic use, but it did not resolve the underlying semantic issues. Invoices remained prone to errors in vendor identification, while contracts highlighted Donut's inability to maintain context across pages. These findings emphasize that while Donut offers greater consistency, OCR+LLM retains flexibility in text-heavy domains, albeit at the cost of stability.

7.5 Limitations of the Experimental Evaluation

While the experimental evaluation provided valuable insights into the comparative performance of Donut and OCR+LLM, several limitations constrain the generalizability of the findings and they must be taken into account when interpreting the results and planning future improvements.

- **Dataset Scope and Coverage**

  The experiments were limited to three domains: invoices, receipts, and government contracts. Although these represent diverse document types, they cannot capture the full variability of real-world document processing tasks. This restricts the extent to which results can be extrapolated to other use cases.

- **Hardware and Computational Constraints**

  Runtime and memory measurements were tied to the available hardware environment. Donut's GPU efficiency came at the cost of high VRAM requirements, while OCR+LLM was constrained by CPU bottlenecks. Moreover, the experiments were conducted on an AWS EC2 g4dn.2xlarge, instance with a single NVIDIA T4 GPU.

This setup, while sufficient for fine-tuning and evaluation, imposed strict limitations on batch size, training duration, and model complexity. As a result, it was not feasible to fine-tune larger models or perform extensive hyperparameter optimization, which may have further improved performance.

- **Time Constraints**

  Due to the academic nature of the project and the finite timeline available, the scope and iteration count of testing were limited. Extended training cycles and comparative evaluations across multiple model variants had to be deprioritized. Time constraints also affected the size of the datasets that could be utilized, since data curation and preparation required substantial manual effort and validation.

8

Conclusion and Future Work

This thesis has presented a comparative study of two approaches for document information extraction: a modular OCR+LLM pipeline and the end-to-end Donut model. While the evaluation provided insights into their respective strengths and limitations, several avenues remain open for further research and system development.

8.1 Conclusion

The experimental results demonstrated that Donut consistently outperformed OCR+LLM in visually complex, short-form documents such as invoices and receipts. Its end-to-end architecture, which jointly models layout, text, and semantics, reduced error propagation and delivered more stable performance in fields that are typically error-prone, such as vendor names and addresses. However, this advantage came with significant hardware requirements, particularly high GPU memory consumption, which may limit applicability in resource-constrained environments.

In contrast, the OCR+LLM pipeline proved more effective for text-heavy, multi-page documents like government contracts. Here, the modular design and ability to process long passages of text provided robustness that Donut could not achieve due to its single-page limitation. Nevertheless, OCR+LLM suffered from higher error variance, as OCR inaccuracies often propagated downstream, limiting its reliability in noisy or heterogeneous document sets.

Beyond accuracy, the study highlighted important practical considerations. Donut offered exceptional processing speed under GPU acceleration, making it suitable for high-volume workflows, while OCR+LLM maintained advantages in modularity and adaptability, allowing components to be upgraded independently. These findings underline that the choice between the two approaches depends not only on performance metrics but also on deployment context, including hardware availability, budget constraints, and document diversity.

The thesis also identified limitations in its scope. Constraints in dataset variety, computational resources, and time restricted the breadth of experiments. Evaluations

relied on standard error metrics (CER, WER), which, while informative, did not fully capture semantic correctness or task-specific utility. These limitations mean the results should be interpreted as indicative rather than definitive.

Overall, the contributions of this work are twofold: (i) a comparative, domain-sensitive evaluation of two contrasting approaches to document information extraction, and (ii) a discussion of the trade-offs that organizations must consider when selecting technologies for intelligent document processing. By highlighting both the technical and practical dimensions of this problem, the thesis offers guidance not only to researchers seeking to improve extraction models but also to practitioners evaluating their deployment feasibility.

In conclusion, while no single approach emerged as universally superior, the analysis underscored the potential of combining strengths from both paradigms. End-to-end models like Donut promise speed and consistency, while modular pipelines retain flexibility and robustness in text-dominant contexts.

8.2 Future Work

This section outlines directions for future work that could enhance both the scientific contribution and the practical applicability of intelligent document processing systems.

8.2.1 Human-in-the-Loop Model Improvement

Integrating human expertise into the extraction pipeline offers a promising strategy for continual model refinement. Instead of treating extraction as a fully automated process, human reviewers could validate or correct selected outputs during routine use. These corrections, if logged systematically, could serve as high-quality supervision signals for iterative fine-tuning. Such a feedback loop would allow the system to adapt to domain-specific nuances, gradually reducing errors in challenging fields such as vendor names with diacritics or lengthy addresses. Research should also investigate efficient annotation interfaces and active learning strategies that prioritize uncertain or high-impact cases, thereby maximizing the benefit of limited human corrections while minimizing reviewer workload.

8.2.2 Retrieval-Augmented Generation

Another promising direction is the integration of retrieval-augmented generation (RAG). In this approach, an extraction model is paired with a retrieval module that supplies external knowledge at inference time. For instance, company identifiers could be cross-validated against an official business registry, or extracted totals could be compared to structured financial records. By grounding predictions in external knowledge sources, RAG has the potential to reduce hallucinations, improve factual accuracy, and ensure compliance with domain constraints. Applied to administrative workflows, RAG could help

bridge the gap between raw document perception and higher-level reasoning, providing outputs that are both contextually accurate and verifiable.

8.2.3 Robustness and Error Mitigation

The error analysis in this thesis identified recurring weaknesses that merit targeted solutions. Errors involving diacritics, domain-specific vendor names, and long addresses suggest that improved normalization techniques should be embedded directly into training. Additional strategies include retrieval-augmented decoding and dictionary-based correction modules. Future research should also focus on improving resilience to OCR noise and typographic variability, as these remain critical barriers to reliable deployment in heterogeneous real-world environments.

8.2.4 Summary

Future research should therefore prioritize three directions: (i) embedding human-in-the-loop mechanisms for adaptive model refinement, (ii) leveraging retrieval-augmented generation to incorporate external knowledge into extraction, and (iii) improving robustness through normalization and error-aware training. Taken together, these efforts would move the field closer to document processing systems that are not only more accurate, but also more transparent, adaptable, and aligned with real-world operational requirements.

Bibliography

[1] F. Ahmed, R. M. Debnath, and M. M. Rahman. "Exploring AI-driven approaches for unstructured document analysis and future horizons". In: *Journal of Big Data* (2023). URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00948-z (cit. on pp. 7, 9).

[2] D. AI. *DeepSeek*. URL: https://www.deepseek.ai (cit. on p. 15).

[3] AiHello. *Creating a Knowledge Graph Without LLMs*. 2023. URL: https://www.aihello.com/resources/blog/creating-a-knowledge-graph-without-llms/ (cit. on p. 16).

[4] Amazon Web Services. *Amazon Textract*. 2025. URL: https://aws.amazon.com/textract/ (cit. on p. 10).

[5] S. Appalaraju et al. "DocFormer: End-to-End Transformer for Document Understanding". In: *arXiv preprint arXiv:2106.11539* (2021). URL: https://arxiv.org/abs/2106.11539 (cit. on pp. 17, 46).

[6] T. B. Brown, B. Mann, N. Ryder, et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2020). URL: https://arxiv.org/abs/2005.14165 (cit. on pp. 11, 15).

[7] S. Choi and Y. Jung. "Knowledge Graph Construction: Extraction, Learning, and Evaluation". In: *Applied Sciences* (2025). URL: https://www.mdpi.com/2076-3417/15/7/3727 (cit. on p. 16).

[8] R. R. Choudhury and S. Dey. "GPT-3 Powered Information Extraction for Building Robust Knowledge Bases". In: *arXiv preprint arXiv:2408.04641* (2023). URL: https://arxiv.org/abs/2408.04641 (cit. on p. 11).

[9] F. Cozzolino. "Document clustering". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2022). URL: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1588 (cit. on p. 9).

[10] F. Cruz. *InvoicesReceiptsPT: Portuguese Invoices and Receipts Dataset*. https://huggingface.co/datasets/Francisco-Cruz/InvoicesReceiptsPT. 2022 (cit. on pp. 3, 4, 22).

[11] D. D. of Data Science. *Label Smoothing: The Overlooked and Underutilized Trick*. 2023. URL: https://blog.dailydoseofds.com/p/label-smoothing-the-overlooked-and (cit. on p. 34).

[12] J. R. Deborah Ferreira. "On the Evaluation of Intelligent Process Automation: A Survey and Research Agenda". In: *arXiv preprint arXiv:2001.02639* (2020). URL: https://arxiv.org/abs/2001.02639 (cit. on p. 14).

[13] J. Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019. URL: https://aclanthology.org/N19-1423/ (cit. on pp. 10, 11).

[14] EdgeDelta. *What Percentage of Data is Unstructured?* Accessed: 2025-09-22. 2023. URL: https://edgedelta.com/company/blog/what-percentage-of-data-is-unstructured (cit. on p. 1).

[15] F. Esposito et al. "Machine Learning for Digital Document Processing: from Layout Analysis to Metadata Extraction". In: *Machine Learning in Document Analysis and Recognition*. Springer, 2008. URL: https://link.springer.com/chapter/10.1007/978-3-540-76280-5_5 (cit. on p. 9).

[16] H. Face. *Donut: Document Understanding Transformer*. Accessed: 2025-09-28. 2025. URL: https://huggingface.co/docs/transformers/en/model_doc/donut (cit. on p. 33).

[17] H. Face. *naver-clova-ix/donut-base*. 2025 (cit. on p. 33).

[18] H. Face. *Trainer — Transformers Documentation*. Accessed: 2025-09-28. 2025. URL: https://huggingface.co/docs/transformers/main_classes/trainer (cit. on p. 33).

[19] H. Face. *Vision Encoder Decoder Models*. Accessed: 2025-09-28. 2025. URL: https://huggingface.co/docs/transformers/model_doc/vision-encoder-decoder (cit. on p. 33).

[20] Forbes. *The Unseen Data Conundrum*. 2022. URL: https://www.forbes.com/councils/forbestechcouncil/2022/02/03/the-unseen-data-conundrum (cit. on p. 1).

[21] Google Cloud. *Cloud Vision API Documentation*. 2025. URL: https://cloud.google.com/vision/docs (cit. on p. 10).

[22] Y. Guo et al. "Normal Workflow and Key Strategies for Data Cleaning: Framework for Real-World Research". In: *Journal of Medical Research* (2023). URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC10557005/ (cit. on p. 18).

[23] S. Huang, J. He, et al. *ICDAR 2019: Scanned Receipts OCR and Information Extraction (SROIE) Dataset*. https://github.com/zzzDavid/ICDAR-2019-SROIE. Accessed: 2025-09-13. 2019 (cit. on pp. 3, 4, 22).

[24] Y. Huang et al. "LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking". In: *arXiv preprint arXiv:2204.08387* (2022). URL: https://arxiv.org/abs/2204.08387 (cit. on pp. 17, 46).

[25] A. Hurst, L. Ahmad, and et al. "GPT-4o System Card". In: *arXiv preprint* arXiv:2410.21276 (2024). URL: https://doi.org/10.48550/arXiv.2410.21276 (cit. on p. 10).

[26] O. Iwashokun and A. Ade-Ibijola. "Parsing of Research Documents into XML Using Formal Grammars". In: *Applied Computational Intelligence and Soft Computing* 2024 (2024), p. 6671359. DOI: 10.1155/2024/6671359. URL: https://doi.org/10.1155/2024/6671359 (cit. on p. 18).

[27] G. Kim et al. "OCR-free Document Understanding Transformer". In: *European Conference on Computer Vision (ECCV)*. 2022. URL: https://arxiv.org/abs/2111.15664 (cit. on pp. 8, 17, 26).

[28] B. Liu. "An Artificial Intelligence Based Approach to Automate Document Processing". MA thesis. Massachusetts Institute of Technology, 2021. URL: https://dspace.mit.edu/handle/1721.1/139571 (cit. on p. 7).

[29] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[30] A. A. M. et al. "Handwritten Text Recognition: A Survey of OCR Techniques". In: *International Journal of Advances in Engineering and Management (IJAEM)* 6.11 (2024), pp. 205–220. ISSN: 2395-5252. DOI: 10.35629/5252-0611205220. URL: https://ijaem.net/issue_dcp/Handwritten%20Text%20Recognition%20%20A%20Survey%20of%20OCR%20Techniques.pdf (cit. on p. 10).

[31] S. V. Mahadevkar et al. "Exploring AI-driven approaches for unstructured document analysis and future horizons". In: *Journal of Big Data* (2024). URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00948-z (cit. on pp. 8, 19).

[32] Microsoft Azure. *Azure AI Vision: AI-powered Image and Video Analysis*. URL: https://azure.microsoft.com/en-gb/products/ai-services/ai-vision/ (cit. on p. 27).

[33] Mindy Support. *A Complete Guide to Semantic Segmentation*. Accessed: 2025-09-27. 2023. URL: https://mindy-support.com/news-post/complete-guide-to-semantic-segmentation/ (cit. on pp. 12, 13).

[34] MIT Sloan. *Tapping the Power of Unstructured Data*. 2022. URL: https://mitsloan.mit.edu/ideas-made-to-matter/tapping-power-unstructured-data (cit. on p. 1).

[35] B. Pfitzmann et al. "DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022. URL: https://dl.acm.org/doi/abs/10.1145/3534678.3539043 (cit. on p. 13).

[36] Portal BASE. *BASE: Contratos Públicos Online*. URL: https://www.base.gov.pt/Base4/pt/pesquisa/?type=contratos (cit. on p. 21).

[37] R. Raj and A. Kos. "A Comprehensive Study of Optical Character Recognition". In: *2022 29th International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*. 2022. URL: https://ieeexplore.ieee.org/document/9837974 (cit. on p. 10).

[38] P. R. Sampaio and H. Maxcici. "Unsupervised Document and Template Clustering using Multimodal Embeddings". In: *arXiv preprint arXiv:2506.12116* (2025). URL: https://arxiv.org/abs/2506.12116 (cit. on p. 9).

[39] Y. Shinyama. *PDFMiner*. Version 20191125. Accessed: 2025-09-28. 2019. URL: https://pypi.org/project/pdfminer/ (cit. on p. 23).

[40] R. Smith. "An overview of the Tesseract OCR engine". In: *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. IEEE. 2007. URL: https://ieeexplore.ieee.org/document/4376991 (cit. on p. 10).

[41] R. K. Tathagata Chakraborti Vatche Isahagian. "From Robotic Process Automation to Intelligent Process Automation". In: (2020). URL: https://www.researchgate.net/publication/344079800_From_Robotic_Process_Automation_to_Intelligent_Process_Automation (cit. on p. 14).

[42] Tesseract Developers. *tesseract-ocr/tesseract (Original GitHub repository)*. https://github.com/tesseract-ocr/tesseract. 2024 (cit. on p. 46).

[43] H. Touvron, T. Lavril, G. Izacard, et al. "LLaMA: Open and Efficient Foundation Language Models". In: *arXiv preprint arXiv:2302.13971* (2023). URL: https://arxiv.org/abs/2302.13971 (cit. on p. 15).

[44] A. Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. 2017. URL: https://arxiv.org/pdf/1706.03762.pdf (cit. on p. 14).

[45] B. Wang et al. *MinerU: An Open-Source Solution for Precise Document Content Extraction*. https://arxiv.org/abs/2409.18839. 2024 (cit. on p. 46).

[46] X.-F. Wang et al. "A Survey of Text Detection and Recognition Algorithms Based on Deep Learning Technology". In: *Neurocomputing* 556 (2023), p. 126702. DOI: 10.1016/j.neucom.2023.126702. URL: https://www.sciencedirect.com/science/article/abs/pii/S0925231223008251 (cit. on p. 10).

[47] Y. Yao. "Analysis for Advancements of Optical Character Recognition in Handwriting Recognition". In: *Proceedings of the 2nd International Conference on Data Analysis and Machine Learning (DAML 2024)*. SciTePress, 2024, pp. 70–74. DOI: 10.5220/0013487600004619. URL: https://www.scitepress.org/Papers/2024/134876/134876.pdf (cit. on p. 10).

[48] M. S. Yeawon Lee Jinseok Son. "BertSRC: transformer-based semantic relation classifications". In: *https://pmc.ncbi.nlm.nih.gov/articles/PMC9446816/* (2022). URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9446816/ (cit. on p. 11).

[49] R. Zargari Marandi, A. S. Frahm, and M. Milojevic. "Datasheets for AI and medical datasets (DAIMS): A data validation and documentation framework before machine learning analysis in medical research". In: *arXiv preprint arXiv:2501.14094* (2025). URL: https://arxiv.org/abs/2501.14094 (cit. on p. 18).

[50] Q. Zhang et al. "Document Parsing Unveiled: Techniques, Challenges, and Prospects for Structured Information Extraction". In: *arXiv preprint arXiv:2410.21169* (2024). URL: https://arxiv.org/abs/2410.21169 (cit. on p. 18).

[51] X. Zhong, J. Tang, and A. J. Yepes. "PubLayNet: Largest Dataset Ever for Document Layout Analysis". In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. 2019. URL: https://arxiv.org/pdf/1908.07836 (cit. on p. 13).